



**UNIVERSITA' DEGLI STUDI DI PADOVA**  
**FACOLTA' DI SCIENZE STATISTICHE**  
**CORSO DI LAUREA IN SCIENZE**  
**STATISTICHE ED ECONOMICHE**

TESI DI LAUREA

**RISCHI OPERATIVI E ALBERI DI GUASTO**  
**ALGORITMI DI ORDINAMENTO**  
**E COMPLESSITA' COMPUTAZIONALE**

RELATORE: CH.MO PROF. MICHELE BONOLLO

LAUREANDA: VALENTINA MAGGIOLO

ANNO ACCADEMICO 2003/2004

*Alla mia famiglia*



## ***RINGRAZIAMENTI***

Ringrazio i miei genitori e mio fratello per aver accettato la mia scelta di continuare gli studi e per aver creduto in me, vi voglio bene.

Ringrazio il mio ragazzo Lorenzo per aver sopportato i miei sbalzi d'umore e per essermi stato sempre vicino.

Ringrazio le mie colleghe e amiche di università, Giò, Cinzia, Miky P. e Miky S., per tutte le volte che mi hanno aiutato e sostenuto, non avrei potuto trovare amiche migliori.

Ringrazio il prof. Michele Bonollo per avermi dato l'opportunità di conoscere un ambiente lavorativo e per quello che in questi mesi ho imparato all'Engineering. E' stata un'esperienza che mi ha fatto crescere e m'ha aiutato a credere nelle mie capacità.

Ringrazio il mio ex-datore di lavoro Massimo Vigo che, per la maggior parte, ha finanziato i miei studi.

Ringrazio tutti i miei amici e amiche, e in particolar modo Chiara, Roberta e Manuela per tutto quello che abbiamo vissuto insieme.



# INDICE

<i>Introduzione</i> .....	1
<i>Capitolo 1</i>	
<b><u>RELAZIONI TRA GLI INSIEMI</u></b> .....	3
1.1 <u>IL LINGUAGGIO DEGLI INSIEMI</u> .....	3
1.1.1 L'insieme.....	3
1.1.2 La coppia ordinata.....	3
1.1.3 Il prodotto cartesiano.....	5
1.1.4 La relazione.....	7
1.1.5 Il dominio e il codominio.....	8
1.1.6 Le funzioni.....	9
1.2 <u>PROPRIETA' DELLE RELAZIONI SU UN INSIEME</u> .....	10
1.3 <u>LA TIPOLOGIA DELLE RELAZIONI</u> .....	12
1.3.1 Relazione Identica.....	12
1.3.2 Relazione di Equivalenza.....	13
1.3.2.1 Classi di equivalenza.....	14
1.3.3 Relazione Inversa.....	16
1.3.4 Relazione d'Ordine.....	17
1.4 <u>UN PREORDINE</u> .....	19
1.5 <u>GRAFI E ALBERI</u> .....	21
1.5.1 Alberi con radice.....	23
1.5.2 Alberi binari.....	26
1.5.2.1 Alberi binari di ricerca.....	28
1.5.3 Alberi N-ari.....	29
1.5.3.1 Trasformazione di alberi N-ari in alberi binari.....	30

1.5.4 Alberi ordinati.....	30
1.6 <u>L'ORDINAMENTO</u> .....	32

## *Capitolo 2*

<b><u>IL RISCHIO OPERATIVO</u></b> .....	35
2.1 <u>DEFINIZIONE DI RISCHIO</u> .....	35
2.2 <u>RISCHIO SPECULATIVO E PURO</u> .....	36
2.3 <u>LE TIPOLOGIE DI RISCHIO</u> .....	37
2.4 <u>IL RISCHIO OPERATIVO IN OTTICA BASILEA 2</u> .....	38
2.4.1 Cos'è il Comitato Basilea 2.....	38
2.4.2 Aspetti problematici di Basilea 2.....	41
2.5 <u>OPERATIONAL RISK</u> .....	43
2.5.1 Metodi di calcolo per i rischi operativi.....	44
2.5.1.1 Metodo dell'indicatore semplice (BIA).....	44
2.5.1.2 Metodo Standard (MS).....	45
2.5.1.3 Metodi avanzati di misurazione (AMA).....	47

## *Capitolo 3*

<b><u>ANALISI TRAMITE ALBERO DI GUASTO</u></b> .....	51
3.1 <u>INTRODUZIONE</u> .....	51
3.2 <u>ENGINEERING INGEGNERIA INFORMATICA SPA</u> .....	52
3.3 <u>IL RISK MANAGEMENT</u> .....	54
3.4 <u>ANALISI DEI RISCHI AZIENDALI</u> .....	54
3.4.1 Fault Tree Analysis.....	55
3.4.2 L'algebra Booleana.....	59
3.4.3 Le fasi di un'analisi Fault Tree.....	61
3.4.3.1 Identificazione del Top Event.....	61
3.4.3.2 Definizione di sistema.....	61
3.4.3.3 Costruzione dell'albero.....	63
3.4.3.4 Valutazione/Analisi dell'albero.....	65

*Capitolo 4*

**APPLICAZIONE DELL'ORDINAMENTO SUGLI ALBERI DI**

**GUASTO**..... 67

4.1 PERCHE' E' IMPORTANTE L'ORDINAMENTO IN UN  
ALBERO DI GUASTO?..... 70

4.2 CREAZIONE DELL'ALGORITMO..... 73

4.2.1 Esempio di un albero di guasto..... 73

4.2.1.1 Calcolo dei livelli di un albero/matrice..... 76

4.2.1.2 Ordinamento dell'albero di guasto..... 77

4.3 COMPLESSITA' COMPUTAZIONALE..... 85

4.3.1 Simboli di Landau..... 86

4.3.2 Analisi Empirica..... 88

***ESERCIZIO COMPLETO***..... 91

***BIBLIO-WEBGRAFIA***.....109





# ***INTRODUZIONE***

Negli ultimi anni la tematica dei *rischi operativi* nella gestione bancaria, che vanno dall'ambito della sicurezza informatica alla possibilità di errori umani nei processi, dal rischio normativo legale a quello di frode e di infedeltà, è stata un terreno fertile per approfondimenti di natura accademica e legata alla prassi aziendale circa l'identificazione, il trattamento e la gestione di questa fattispecie di rischio. La proposta di Basilea 2 propone per la prima volta l'inclusione di tali rischi nell'ambito della gamma dei rischi che le banche devono prendere in considerazione nella determinazione della copertura patrimoniale utile ai fini di vigilanza.

I rischi operativi, ovviamente, sono sempre esistiti, pur non essendo mai stati individuati quale categoria a sé stante, meritevole pertanto di un autonomo trattamento come i rischi finanziari.

Ma nel comparto industriale, i rischi operativi sono percepiti come elementi chiave del *risk management* e, le esigenze di profitto e di competitività hanno posto da più tempo il problema di un adeguato trattamento degli stessi.

Grande merito va, pertanto, a Basilea 2 per la rilevanza degli sforzi compiuti nel circoscrivere le fattispecie di rischi operativi, in quanto categoria eterogenea e non meramente o necessariamente residuale rispetto ai rischi di credito e di mercato nell'ambito della gestione degli intermediari finanziari.

Questa tesi è il lavoro di alcuni mesi di stage presso la società Engineering Ingegneria Informatica SPA nell'ambito di un progetto informatico per l'ordinamento di alberi di guasto.

*Il capitolo 1* spiegherà diversi concetti legati:

- Agli insiemi, alle relazioni che esistono tra essi, e in particolar modo alla relazione d'ordine specificando cosa s'intende per preordine;
- Al concetto di albero, alle tipologie di alberi e al loro ordinamento;

Il capitolo 2 mirerà a presentare il rischio nelle sue diversità, soffermandosi sui Rischi Operativi legati al concetto di Comitato Basilea 2.

Il capitolo 3 tratterà dell'analisi tramite albero di guasto, presentando:

- Engineering SPA e le sue attività principali;
- Fault Tree Analysis e le sue fasi di sviluppo;

Il capitolo 4 analizzerà in tutte le sue parti l'algoritmo creato in Mathcad, fornendo un esempio pratico e calcolando la complessità computazionale empirica.

## *CAPITOLO 1*

# RELAZIONI TRA GLI INSIEMI

### 1.1 IL LINGUAGGIO DEGLI INSIEMI

#### 1.1.1 L'INSIEME

Dal dizionario<sup>1</sup>: Insieme = complesso, totalità di persone o cose;

Tale definizione non è matematicamente corretta perché non dà un criterio che permetta di decidere se un elemento appartiene oppure no all'insieme.

Definizione di insieme secondo il matematico Cantor<sup>2</sup>: insieme è una collezione di oggetti qualsiasi, ben definiti e distinguibili, che fanno parte della nostra intuizione e del nostro pensiero.

Tale definizione è matematicamente corretta perché i termini "ben definiti e distinguibili" rappresentano implicitamente un criterio di scelta.

In effetti, dal punto di vista formale, i termini insieme ed elemento sono termini primitivi che possono essere definiti ricorrendo a sinonimi come aggregato, collezione, ecc.

Perciò considereremo insieme un qualunque aggregato (o collezione) di oggetti per il quale sia sempre possibile decidere se un generico oggetto appartiene oppure no all'aggregato stesso.

Indicheremo gli insiemi con le lettere maiuscole dell'alfabeto.

#### 1.1.2 LA COPPIA ORDINATA

Il concetto di *coppia ordinata* è uno di quelli di più largo consumo in matematica. Intuitivamente, dati due elementi  $a$  e  $b$  non necessariamente distinti,

---

<sup>1</sup> Il Dizionario della Lingua Italiana di G. Devoto e G.C. Oli – Casa Editrice Le Monnier.

<sup>2</sup> Georg Cantor (Pietroburgo 1845, Halle 1918) matematico russo a cui dobbiamo i più importanti contributi nel campo delle serie trigonometriche, sui numeri reali come insieme non enumerabile, sulla teoria delle dimensioni e, soprattutto, alcuni fondamentali risultati sulla teoria degli insiemi.

la coppia  $(a,b)$  consiste negli elementi "presi in questo ordine", cioè con  $a$  "primo elemento" e  $b$  "secondo elemento". Per una definizione rigorosa occorrerebbe avere già la nozione di ordine, che invece è sempre successiva a quella di coppia ordinata. Si può ovviare a questa difficoltà definendo una coppia nel seguente modo:

$$(a,b) = \{\{a\}, \{a,b\}\}$$

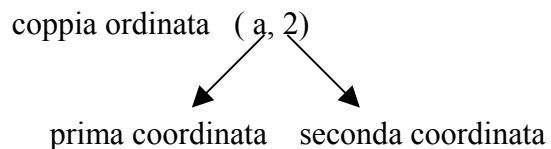
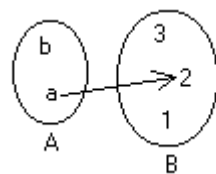
E' molto importante osservare che  $(a,b)$  è diverso da  $\{a,b\}$ . La cosa è evidente in particolare se  $a$  e  $b$  sono uguali, nel qual caso l'insieme  $\{a,b\}$  coincide con l'insieme  $\{a\}$ , mentre  $(a,b)$  coincide, secondo la definizione che abbiamo dato, con  $\{\{a\}\}$ .

Facciamo un esempio:

Consideriamo l'insieme  $A = \{a, b\}$  e l'insieme  $B = \{1, 2, 3\}$ . Se prendiamo un elemento di  $A$ , per esempio  $a$ , ed un elemento di  $B$ , per esempio  $2$ , possiamo costruire la **coppia ordinata** :

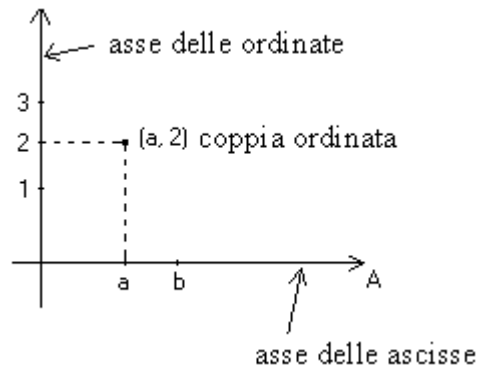
$$(a,2)$$

dove è **essenziale l'ordine** con cui si scelgono gli elementi dai due insiemi. Il primo elemento della coppia ordinata, quello scritto a sinistra, si chiama **prima coordinata** mentre il secondo, quello scritto a destra, si chiama **seconda coordinata**:



Nei diagrammi di Eulero<sup>3</sup>-Venn<sup>4</sup> una coppia ordinata viene rappresentata da una **freccia** che parte dalla prima coordinata della coppia ordinata e punta alla seconda coordinata della medesima.

Vi è un altro modo molto proficuo di rappresentare le coppie ordinate utilizzando gli **assi cartesiani**



Sugli assi cartesiani una coppia ordinata viene rappresentata con un **punto** come illustrato in figura.

Utilizzando gli assi cartesiani occorre sottolineare che l'insieme da cui si prendono le prime coordinate va posto sull'asse delle **ascisse** (l'asse **orizzontale**) mentre l'altro insieme, da cui si prendono le seconde coordinate, va posto sull'asse delle **ordinate** (l'asse **verticale**).

### 1.1.3 IL PRODOTTO CARTESIANO

Dati due insiemi  $A$  e  $B$ , il loro *prodotto cartesiano*, indicato con  $A \times B$ , è costituito da tutte e sole le coppie ordinate  $(a, b)$ , in cui  $a \in A$  e  $b \in B$ .

A proposito del prodotto cartesiano valgono i seguenti risultati:

<sup>3</sup> Leonhard Eulero (Basilea 1707, San Pietroburgo 1783) astronomo e filosofo svizzero. La sistematizzazione e la riformulazione dell'analisi che si trova nelle sue opere è alla base della matematica moderna e della teoria delle funzioni.

<sup>4</sup> John Venn (Drypool 1834, Cambridge 1923) logico inglese. A lui va ricondotto il concetto di "logica simbolica", che è anche il titolo della sua opera principale, in cui approfondisce i rapporti tra logica, linguaggio e pensiero comune.

- Per ogni insieme  $A$  si ha  $A \times \emptyset = \emptyset \times A = \emptyset$ .
- Se  $A$  e  $B$  sono insiemi non vuoti si ha  $A \times B = B \times A \Leftrightarrow A = B$ .

Il prodotto cartesiano  $A \times B$  è definito allora da :

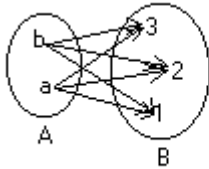
$$A \times B = \{(a,b); a \in A, b \in B\}$$

che si legge " il prodotto cartesiano dell'insieme  $A$  per l'insieme  $B$  è l'insieme di tutte le coppie ordinate che si ottengono prendendo la prima coordinata in  $A$  e la seconda coordinata in  $B$  ".

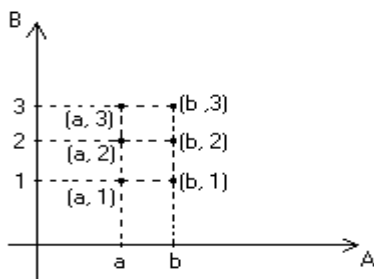
Considerando gli insiemi  $A$  e  $B$  definiti precedentemente si ha allora :

$$A \times B = \{(a,1), (a,2), (a,3), (b,1), (b,2), (b,3)\}$$

Graficamente:



Prodotto cartesiano  $A \times B$  usando i diagrammi di Eulero-Venn ovvero prendendo tutte le possibili frecce dagli elementi da  $A$  agli elementi di  $B$



Prodotto cartesiano  $A \times B$  usando gli assi cartesiani dove si vede bene che le coppie ordinate del prodotto cartesiano sono indicate da **tutti i possibili punti** che si possono ottenere considerando gli elementi dei due insiemi

### 1.1.4 LA RELAZIONE

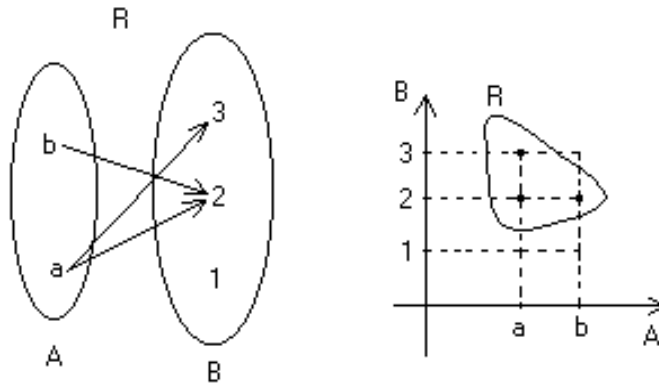
In matematica, il concetto di **relazione** è analogo a quello del linguaggio comune. Vi è una relazione quando elementi di un insieme sono **legati** in qualche modo con elementi di un altro insieme. Gli elementi di due insiemi possono essere di **qualunque tipo** ed il legame fra loro può essere di qualsiasi natura.

Quindi una **relazione binaria** fra due insiemi, in matematica, è semplicemente un **sottoinsieme del prodotto cartesiano** fra due insiemi, di conseguenza anch'esso un **insieme**, ovvero un oggetto del tutto definito, per il quale non è possibile alcuna ambiguità ed imprecisione.

Per esempio, rispetto agli insiemi  $A$  e  $B$  degli esempi precedenti, possiamo definire la relazione :

$$R = \{(a,2), (a,3), (b,2)\}$$

che è palesemente un sottoinsieme di  $A \times B$  e la possiamo visualizzare nei due modi :

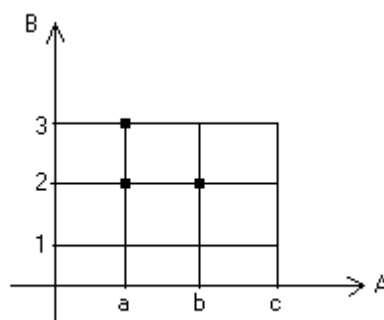




### 1.1.5 IL DOMINIO E IL CODOMINIO

Consideriamo gli insiemi  $A = \{a, b, c\}$  e  $B = \{1, 2, 3\}$ . Costruiamo fra essi la relazione :

$$R = \{(a,2), (a,3), (b,2)\}$$



Chiameremo **dominio** di una relazione l'insieme degli elementi del primo insieme che sono "coinvolti" nella relazione. Denominiamo il dominio di  $R$  col simbolo  $D(R)$ .

Chiameremo **codominio** di una relazione l'insieme degli elementi del secondo insieme che sono "coinvolti" nella relazione. Denominiamo il codominio di  $R$  col simbolo  $C(R)$ .

Nell'esempio considerato sopra avremo :

$$D(R) = \{a,b\} \text{ e } C(R) = \{2,3\}$$

Il dominio ed il codominio di una relazione sono quindi sottoinsiemi rispettivamente del primo e del secondo insieme su cui è costituita la relazione. In particolare il dominio ed il codominio **possono coincidere** con il primo ed il secondo insieme (rispettivamente).

### 1.1.6 LE FUNZIONI

Una funzione da un insieme  $A$  a un insieme  $B$  è una corrispondenza da  $A$  a  $B$  tale che ad ogni elemento di  $A$  associa uno ed un solo elemento di  $B$ .

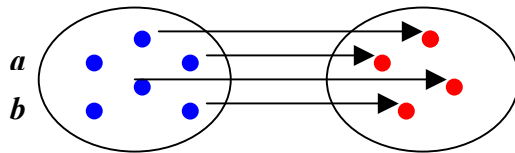
Una funzione  $f$  fra due insiemi  $A$  e  $B$  soddisfa le seguenti proprietà :

$$D(f) = A$$

$$((x, y) \in f, (x, z) \in f) \Rightarrow (y = z)$$

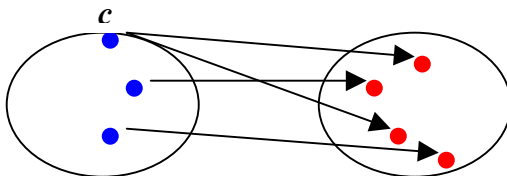
Facciamo degli esempi, nominiamo l'insieme di sinistra  $A$  e l'insieme di destra  $B$ :

caso a)



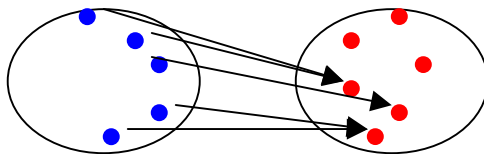
Nel caso a) la corrispondenza non è una funzione perché agli elementi  $a$  e  $b$  dell'insieme di sinistra  $A$  non corrisponde nulla.

caso b)



Nel caso b) la corrispondenza non è una funzione perché all'elemento  $c$  dell'insieme di sinistra  $A$  corrispondono due elementi.

caso c)



Nel caso c) la corrispondenza è una funzione perché ad ogni punto dell'insieme di sinistra  $A$  parte una e una sola freccia.

Facciamo inoltre queste distinzioni:

- Una funzione  $f$  da  $A$  a  $B$  si dice *iniettiva* se ad elementi distinti di  $A$  fa corrispondere elementi distinti di  $B$ .
- Una funzione da  $A$  a  $B$  si dice *suriettiva* se ogni elemento di  $B$  corrisponde a qualche elemento di  $A$  tramite  $f$ .
- L'immagine di una funzione  $f$  da  $A$  a  $B$  è l'insieme degli elementi del tipo  $f(a)$  per qualche  $a$  in  $A$  (insieme di punti di arrivo di frecce).
- Se una funzione  $f$  è sia *iniettiva* che *suriettiva* allora si dice che è una *biiezione*.
- Se  $f$  è una funzione da  $A$  a  $B$  e  $g$  è una funzione da  $B$  a  $C$ , la funzione composta  $g \circ f$  è la funzione da  $A$  a  $C$  ottenuta applicando a ciascun elemento  $a$  di  $A$  la funzione  $f$  e poi applicando la funzione  $g$  al risultato  $f(a)$ .

## 1.2 PROPRIETA' DELLE RELAZIONI SU UN INSIEME

Le relazioni possono godere di alcune proprietà che sono particolarmente importanti nelle applicazioni. Considerando che la condizione  $a \in A$  è sempre sottintesa e che le relazioni considerate sono non vuote, elenchiamo le proprietà fondamentali:

### Assioma di riflessività

Una relazione  $R$  su un insieme  $A$  è:

- *riflessiva* se  $\forall a \in A (a, a) \in R$ ;
- *irriflessiva* se  $\forall a \in A (a, a) \notin R$ ;

**Assioma di simmetria**

Una relazione **R** su un insieme  $A$  è:

- *simmetrica* se  $\forall a,b \in A (a,b) \in R \Rightarrow (b,a) \in R$ ;
- *asimmetrica/non simmetrica* se  $\forall a,b \in A (a,b) \in R \Rightarrow (b,a) \notin R$ ;
- *antisimmetrica* se  $\forall a,b \in A ((a,b) \in R \wedge (b,a) \in R) \Rightarrow a=b$ ;

**Assioma di transitività**

Una relazione **R** su un insieme  $A$  è:

- *transitiva* se  $\forall a,b,c \in A ((a,b) \in R \wedge (b,c) \in R) \Rightarrow (a,c) \in R$ ;
- *non transitiva* se  $\forall a,b,c \in A ((a,b) \in R \wedge (b,c) \in R) \Rightarrow (a,c) \notin R$ ;

**Assioma di completezza**

Una relazione **R** su un insieme  $A$  è:

- *completa* se  $\forall a,b \in A (a,b) \notin R \Rightarrow (b,a) \in R$ ;
- *quasi completa* se  $\forall a,b \in A a \neq b \wedge (a,b) \notin R \Rightarrow (b,a) \in R$ ;

**Proposizioni elementari:**

- Asimmetria  $\Rightarrow$  Irriflessività;
- Irriflessività e Transitività  $\Rightarrow$  Asimmetria,
- Completezza  $\Rightarrow$  Riflessività;
- Asimmetria e Non Transitività  $\Rightarrow$  Transitività e Irriflessività;

Facciamo un semplice esempio per capire meglio queste proprietà:

dato l'insieme  $A = \{a,b,c,d,e,f\}$  siano  $R_1 = \{(a,f),(b,a),(c,d),(d,d),(e,a),(e,b),(f,c)\}$  e  $R_2 = \{(a,a),(b,b),(c,c),(d,d),(e,e),(f,f)\}$  due relazioni.

La prima relazione,  $R_1$ :

- non è riflessiva, in quanto ad esempio  $(a,a) \notin R_1$ ;

- non è simmetrica, in quanto ad esempio  $(a,f) \in R_1$  ma  $(f,a) \notin R_1$ ;
- è antisimmetrica, in quanto ogni punto non contiene anche il simmetrico, ma a differenza dell'asimmetria può contenere una parte o tutta la bisettrice stessa;
- non è transitiva, in quanto ad esempio  $(a,b) \in R_1$  e  $(a,f) \in R_1$  ma  $(b,f) \notin R_1$ ;

La seconda relazione,  $R_2$ :

- è riflessiva;
- è simmetrica;
- è antisimmetrica;
- è transitiva;

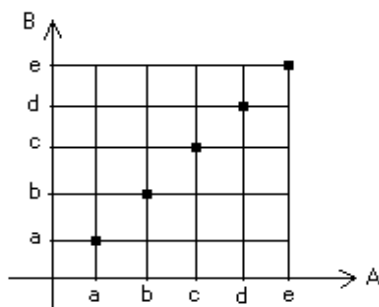
## 1.3 LA TIPOLOGIA DELLE RELAZIONI

### 1.3.1 RELAZIONE IDENTICA

Se consideriamo una relazione fra un insieme e se stesso per cui ad ogni elemento corrisponda se stesso, otteniamo una relazione particolarmente importante, la cosiddetta **relazione identica**.

Per esempio, se  $A = \{a, b, c, d, e\}$ , la relazione identica in  $A$  (ovvero da  $A$  ad  $A$ ) è :

$$I = \{(a,a), (b,b), (c,c), (d,d), (e,e)\}$$



Si noti la particolare forma grafica a "diagonale" di una relazione identica.

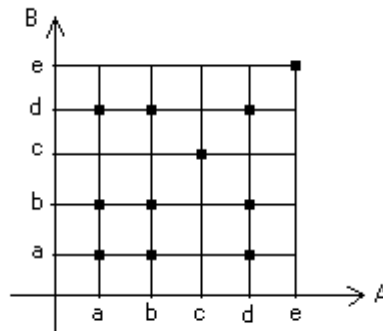
### 1.3.2 RELAZIONE DI EQUIVALENZA

Consideriamo un insieme formato da cinque amici:

$$A = \{a, b, c, d, e\}$$

Studiamo la relazione:  $R = \text{“}a, b, d \text{ sono fratelli”}$  all'interno dell'insieme  $A$  dei cinque amici.

Rappresentiamo graficamente la relazione aggiungendo il fatto che ogni amico può essere considerato **fratello di se stesso** (in matematica si fanno spesso "bizzarre" asserzioni, tipo questa, che però non intaccano la logica, ma possono essere utili a creare opportune generalizzazioni) :



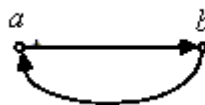
Si vede bene che questa relazione soddisfa tre importanti **proprietà** :

- proprietà **riflessiva** : ogni elemento è in relazione con se stesso.

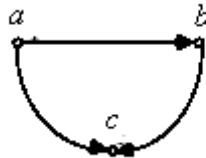
Ovvero:  $a R a$ ;



- proprietà **simmetrica** : se un elemento è in relazione con un altro, allora il secondo è in relazione col primo. Ovvero :  $a R b \Rightarrow b R a$ ;



- proprietà **transitiva** : se un elemento è in relazione con un secondo elemento ed il secondo elemento è in relazione con un terzo elemento, allora il primo elemento è in relazione col terzo elemento. Ovvero :  $a R b$ ,  $b R c \Rightarrow a R c$ ;



Quindi possiamo dire che tutte le relazioni fra un insieme e se stesso che soddisfano le tre proprietà definite sopra si chiamano **relazioni di equivalenza** e rappresentano un tipo di relazione di fondamentale importanza per tutta la matematica.

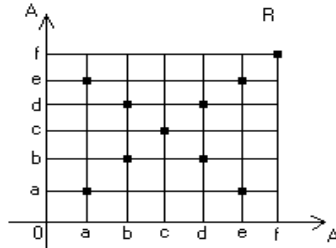
### 1.3.2.1 CLASSI DI EQUIVALENZA

Consideriamo l'insieme di alcuni amici. Chiameremo  $A$  questo insieme ed indicheremo con le lettere minuscole i singoli amici. Supponiamo che sia  $A = \{a, b, c, d, e, f\}$ . Supponiamo inoltre che :

AMICI	ANNI
$a$	15
$b$	17
$c$	20
$d$	17
$e$	15
$f$	50

Creiamo la relazione  $ARA$  definita dall'affermazione "avere la stessa età".

Graficamente :



Supponendo che ogni amico abbia la stessa età di se stesso, la relazione  $R$  è evidentemente una relazione di equivalenza perché soddisfa le proprietà riflessiva, simmetrica e transitiva.

Consideriamo ora un elemento dell'insieme  $A$ , per esempio  $a$ , e costruiamo l'insieme degli elementi in relazione con esso (cioè equivalenti ad esso). Indicando questo insieme col simbolo  $[a]$  otterremo allora:

$$[a] = \{a, e\}$$

Questo insieme si chiama **classe di equivalenza** di  $a$ .

Facendo la stessa cosa per gli altri elementi di  $A$  (e ripetendo ancora  $[a]$ ) otteniamo:

$$[a] = \{a, e\}$$

$$[b] = \{b, d\}$$

$$[c] = \{c\}$$

$$[d] = \{b, d\}$$

$$[e] = \{a, e\}$$

$$[f] = \{f\}$$

La definizione matematica di classe di equivalenza è allora :

$$[a] = \{x, x \in A, (a, x) \in R\}$$

Questa nozione è strettamente collegata a quella di partizione. Una classe  $C$  di sottoinsiemi di  $A$  è detta una **partizione** di  $A$  se:

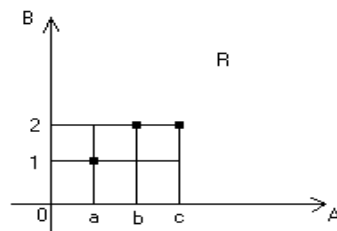


- gli elementi di  $C$  sono a due a due disgiunti;
- l'unione degli elementi di  $C$  coincide con  $A$ .

Gli elementi di  $C$  vengono anche chiamati *classi*. In altre parole  $C$  è una partizione se rappresenta un modo di dividere gli elementi di  $A$  in tante parti separate. Ad esempio se  $A$  è l'insieme dei numeri interi,  $P$  è l'insieme dei numeri pari e  $D$  è l'insieme dei numeri dispari, allora una partizione di  $A$  si ottiene ponendo  $C = \{P, D\}$ .

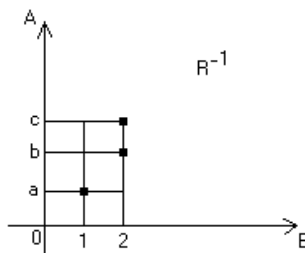
### 1.3.3 RELAZIONE INVERSA

Consideriamo la seguente **relazione**  $ARB$  :



Da questa relazione è possibile costruire la **relazione inversa**  $BR^{-1}A$  semplicemente invertendo le coppie ordinate. La coppia  $(a, 1)$  diventa  $(1, a)$  ecc.

Così facendo si inverte il dominio con il condominio, si ha cioè :



La definizione matematica esatta di relazione inversa è :

$$\mathbf{R^{-1}} = \{(b, a); b \in B, a \in A, (a, b) \in R\}$$

### 1.3.4 RELAZIONE D'ORDINE

Diremo relazione d'ordine una relazione che sia transitiva e, in aggiunta:

- riflessiva e antisimmetrica (**relazione d'ordine larga/parziale** “ $\succsim$ ”)
- irreflessiva e asimmetrica (**relazione d'ordine stretta** “ $\succ$ ”).

Una **relazione d'ordine larga/parziale** nell'insieme  $A = \{a, b, c, \dots\}$  è tale se:

- $a$  è in relazione con  $b$  e  $b$  è in relazione con  $c$  allora  $a$  è in relazione con  $c$ , cioè  $((a,b) \in R, (b,c) \in R) \Rightarrow ((a,c) \in R)$ ;
- ogni elemento  $a$  appartenente ad  $A$  è in relazione con se stesso, cioè  $(a,a) \in R$ ;
- $a$  è in relazione con  $b$  e  $b$  è in relazione con  $a$  allora  $a$  è uguale a  $b$  e viceversa, cioè  $((a,b) \in R, (b,a) \in R) \Rightarrow (a=b)$ ;

Una **relazione d'ordine stretta** nell'insieme  $A = \{a, b, c, \dots\}$  è tale se:

- $a$  è in relazione con  $b$  e  $b$  è in relazione con  $c$  allora  $a$  è in relazione con  $c$ , cioè  $((a,b) \in R, (b,c) \in R) \Rightarrow ((a,c) \in R)$ ;
- ogni elemento  $a$  appartenente ad  $A$  non è in relazione con se stesso, cioè  $(a,a) \notin R$ ;
- $a$  è diverso da  $b$  allora o  $a$  è in relazione con  $b$  oppure  $b$  è in relazione con  $a$ , cioè  $(a,b \in A, a \neq b) \Rightarrow ((a,b) \in R \text{ oppure } (b,a) \in R)$ ;

**Un ordine totale (o lineare)** è una relazione d'ordine che gode della proprietà transitiva, riflessiva, antisimmetrica e di completezza quindi un ordine totale è contemporaneamente un ordine parziale.

Facciamo un esempio:

Considerato che l'insieme  $A$  è composto solo da numeri interi diversi da zero, si deduce immediatamente che la relazione  $R$ , così definita

$$\forall (a,b) \in A \times A, aRb \Leftrightarrow a \text{ divide } b$$

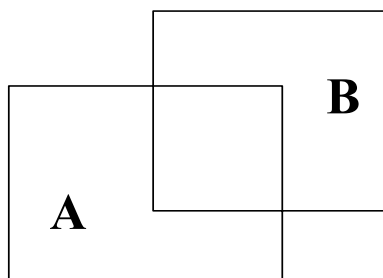
gode delle seguenti proprietà:

- **riflessiva** infatti ogni numero è divisore di se stesso;
- **transitiva** se  $a$  è divisore di  $b$  ed  $b$  è divisore di  $c$  allora anche  $a$  è divisore di  $c$ ;
- **antisimmetrica** se  $a$  è divisore di  $b$  e  $b$  è divisore di  $a$  allora  $a$  e  $b$  sono uguali.
- Si noti, comunque che **la relazione d'ordine nell'insieme è parziale**. Infatti esistono coppie di elementi  $(a,b)$  per i quali non si verifica che  $a$  sia divisore di  $b$ , né che  $b$  sia divisore di  $a$ .

Ad esempio  $(2,5)$ ,  $(4,5)$ ,  $(5,8)$ ,...

Facciamo un altro esempio:

I rettangoli  $A$  e  $B$  nella figura seguente rappresentano due insiemi nessuno dei quali è contenuto nell'altro



Quindi l'inclusione tra i due insiemi è una relazione riflessiva, antisimmetrica e transitiva, ma non completa. Essa è quindi un ordine parziale ma non totale.

Le relazioni oltre ad essere binarie possono anche essere  $n$ -arie e quest'ultime sono alla base dei linguaggi che trattano i data-base relazionali. Ad esempio la seguente tabella

NOME	COGNOME	ETA'	SESSO	STIPENDIO
Marco	Schiraldi	40	M	2.000,00
Anna	Bianchi	35	F	1.500,00
Lorenzo	Zambonin	25	M	1.500,00

è tipica dei linguaggi relazionali, ed è una relazione costituita da tre 5-tuple, cioè è un sottoinsieme del prodotto cartesiano

$$\text{Nome} \times \text{Cognome} \times \text{Età} \times \text{Sesso} \times \text{Stipendio}$$

In questo caso *Nome* è l'insieme dei possibili nomi, *Cognome* l'insieme dei possibili cognomi e così via. I linguaggi relazionali invece di fare calcoli con numeri fanno calcoli con tabelle di questo tipo, cioè sono un calcolo delle relazioni.

## 1.4 UN PREORDINE

Una relazione  $\succsim$  che sia riflessiva e transitiva viene detta: preordine.

Se la relazione  $\succsim$  è anche completa viene detta preordine totale.

Si noti che il termine “totale” implica “riflessivo”. Quindi un preordine totale può essere sinteticamente detto relazione transitiva e totale.

Enunciamo ora tre teoremi:

### TEOREMA 1:

*Sia  $X$  un insieme e  $\succsim$  un preordine totale. Allora  $\succ$ , definita così:  $x \succ y \Leftrightarrow x \succsim y$*

*e non  $(y \succsim x)$ , è:*

- *asimmetrica cioè  $\nexists x, y \in X$  tali che  $(x \succ y \text{ e } y \succ x)$*
- *negativamente transitiva cioè  $\forall x, y, z \in X (x \succ y \Rightarrow (x \succ z \text{ oppure } z \succ y))$*

### TEOREMA 2:

*Sia  $X$  un insieme e sia  $\succ$  una relazione asimmetrica e negativamente transitiva.*

*Allora  $\succsim$ , definita così:  $x \succsim y \Leftrightarrow \text{non } (y \succ x)$ , è un preordine totale.*

**TEOREMA 3:**

Sia  $X$  un insieme e sia  $\succcurlyeq$  un preordine totale su  $X$ . Allora la relazione  $\sim$ , definita così:  $x \sim y \Leftrightarrow (x \succcurlyeq y \text{ e } y \succcurlyeq x)$ , risulta essere una relazione di equivalenza.

Quindi possiamo affermare che vi sono legami fra  $\succcurlyeq$ ,  $\succ$  e  $\sim$ .

- Dato  $X$  e  $\succcurlyeq$  possiamo definire  $\succ$  e  $\sim$  nel modo seguente:

$$x \succ y \Leftrightarrow x \succcurlyeq y \text{ e non } (y \succcurlyeq x)$$

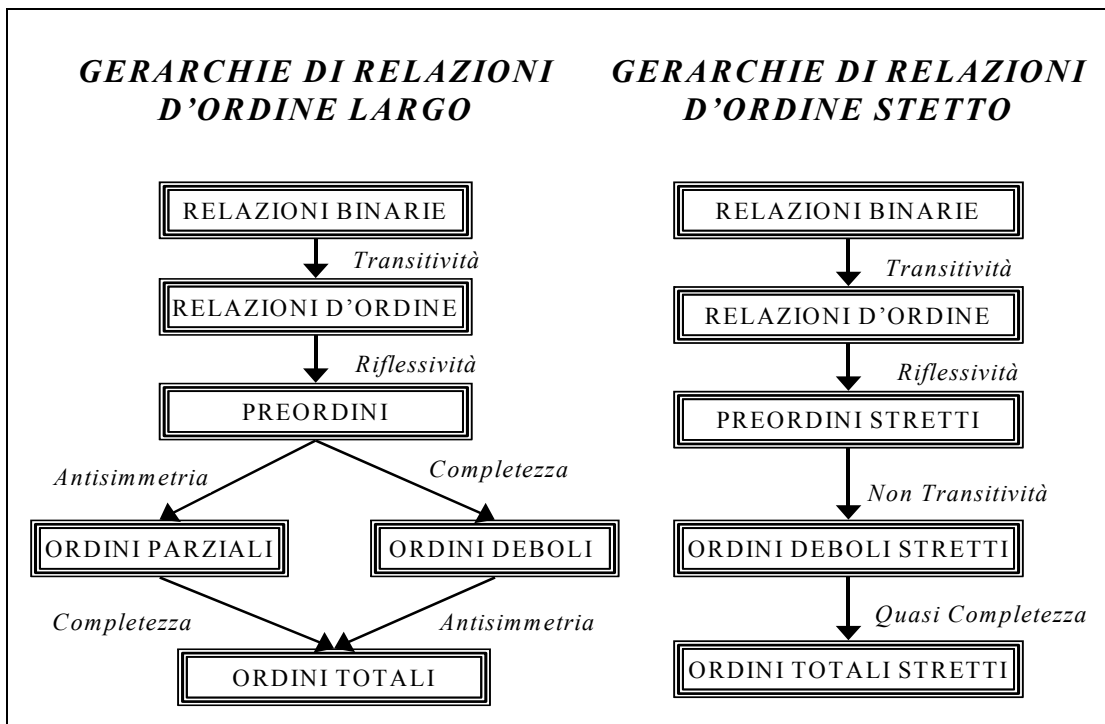
$$x \sim y \Leftrightarrow (x \succcurlyeq y \text{ e } y \succcurlyeq x)$$

- Dato  $X$  e  $\succ$  possiamo definire  $\succcurlyeq$  nel modo seguente:

$$x \succcurlyeq y \Leftrightarrow \text{non } (y \succ x) \text{ e poi ovviamente si può arrivare a } \sim.$$

E' anche abbastanza ovvio come da  $\sim$  non si possa giungere a  $\succcurlyeq$  oppure a  $\succ$ .

Facciamo ora uno schema riassuntivo:



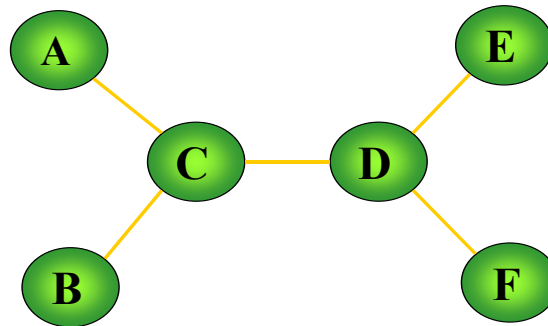
## 1.5 GRAFI E ALBERI

Quando  $A$  è un insieme finito, una relazione binaria in  $A$  viene a volte chiamata **grafo orientato** o, semplicemente **grafo**.

I **grafi** sono delle strutture matematiche costituite da **nodi**, elementi di un insieme singolarmente individuabili con un **etichetta** o una chiave, e **spigoli** o connessioni che connettono i nodi a coppie.

Un grafo non orientato e senza cicli è detto **foresta**; se per di più esso è connesso allora viene detto **albero**. Un albero è un insieme di elementi (detti nodi) collegati tra loro da una relazione d'ordine parziale; la relazione è tale che ogni nodo ha uno e un solo nodo che lo precede (detto il suo *padre*), con l'eccezione di un unico nodo che non ha padre e che precede tutti gli altri nodi, detto *radice*. Ogni foresta è un insieme di alberi, in quanto ogni componente connessa di una foresta è un albero.

Il grafo sotto riportato è una foresta composta da 2 alberi:

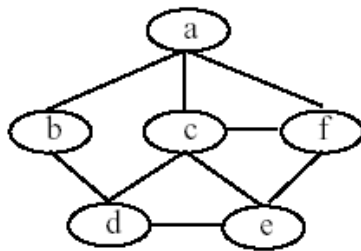


In un grafo una **catena** fra due nodi è una sequenza di spigoli che è necessario percorrere per raggiungere uno dei due nodi partendo dall'altro. Ad esempio: se esiste uno spigolo tra il nodo A e il nodo C, ed uno spigolo tra il nodo B e il nodo C, esiste una catena che congiunge il nodo A con il nodo B che passa per il nodo C ed è formato da questi due spigoli.

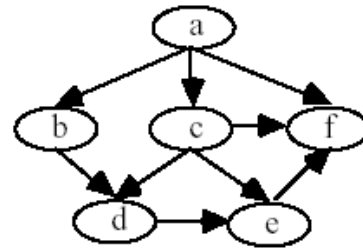
In un grafo possono esistere più catene che congiungono due nodi, le catene minime sono quelle con il minor numero di spigoli possibile.

Nella definizione di albero imporre che i figli non siano connessi fra di loro e che siano connessi al loro padre con un solo spigolo è equivalente ad imporre che la catena più breve che congiunge fra loro due nodi, figli dello stesso padre, sia formata da due spigoli. I grafi possono avere spigoli orientati (detti archi), cioè con un nodo di partenza e uno di arrivo. Questo tipo di grafo viene detto **grafo orientato** o **digrafo** (**directed graph** in inglese).

Se rinunciamo al fatto che esista un unico spigolo fra figlio e padre otteniamo una nuova famiglia di alberi che vengono detti **grafi aciclici**. I **grafi orientati aciclici** (DAG direct acyclic graph) sono strutture fondamentali in linguistica.



a) grafo non orientato

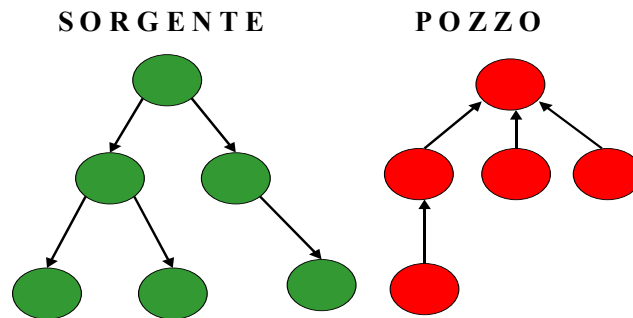


b) grafo orientato

Un albero è libero se è un grafo non orientato, aciclico e connesso. L'albero è detto libero per ribadire che non vi è un nodo prefissato come radice, né il conseguente, implicito, orientamento dei rami.

Importante è ricordare che se l'albero è un grafo non orientato allora qualsiasi nodo può considerarsi radice ma se l'albero è orientato bisogna distinguere due casi:

- La radice ha solo archi in uscita (albero sorgente);
- La radice ha solo archi in entrata (albero pozzo);



Ricapitolando un insieme  $A$  di elementi si dice albero se:

- $A = \emptyset$ ;
- oppure:
- $A = A(r, A_0, A_1, \dots, A_{n-1})$ ;  $r = \text{radice}$ ;  
 $A_i = \text{albero, detto "sottoalbero } i\text{-esimo di } A"$ ;

Definizione ricorsiva: struttura dati vuota, oppure costituita da un elemento detto radice e da altri elementi che formano insiemi disgiunti chiamati sottoalberi della radice, ciascuno dei quali è ancora un albero;

### 1.5.1 ALBERI CON RADICE

Per passare da un albero libero ad un albero radicato basta scegliere, semplicemente, un nodo come radice e si può ben dire che questo cambia radicalmente le cose... Infatti possiamo ora considerare tutti e soli i cammini che portano dalla radice e ordinare i nodi gerarchicamente secondo la loro distanza dalla radice stessa.

Array, liste concatenate, pile, code sono strutture dati lineari ed è possibile utilizzarle per una rappresentazione gerarchica dei dati; questo è possibile con una struttura dati chiamata albero.

Qual è lo scopo di una rappresentazione gerarchica di dati? Ovvero: perché usare gli alberi?

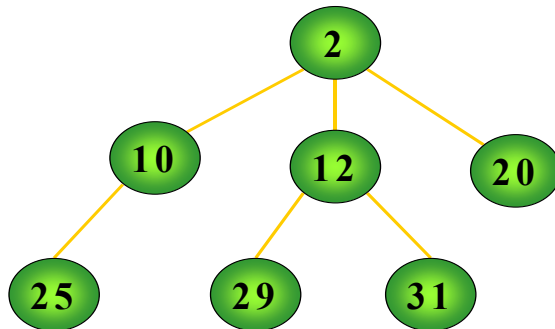


Le operazioni di ricerca, inserimento e cancellazione sulle strutture dati lineari hanno un tempo lineare, su una struttura gerarchica come un albero possono essere rese più efficienti.

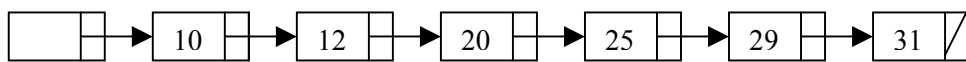
Consideriamo un insieme di numeri naturali:

[2, 10, 12, 20, 25, 29, 31]

Lo rappresentiamo con struttura gerarchica ad albero:



E con struttura dati lineare lista concatenata:



otteniamo una lista in cui il primo elemento è la radice dell'albero e gli elementi successivi sono le liste che rappresentano i sottoalberi della radice stessa:

[2,[10,[25]], [12,[29],[31]], [20]]

oppure utilizzando liste di uguale lunghezza dove l'atomo "NULL" rappresenta l'albero vuoto:

[2, [10, [25,NULL,NULL,NULL], NULL,NULL], [12, [29,NULL,NULL,NULL], [31,NULL,NULL,NULL], NULL], [20,NULL,NULL,NULL]]

Gli **alberi** sono una famiglia particolare di grafi molto importante e possiedono una terminologia specifica e spesso diversa da quella che viene usata comunemente nella teoria dei grafi.

In un albero esiste un nodo privilegiato che viene chiamato **radice**.

I nodi connessi alla radice con un ramo vengono chiamati **figli** e la radice il loro **padre**. Se per esempio  $u$  è il vertice di un albero radicato  $A$  ed  $r$  è la sua radice, allora ogni vertice  $v$  sull'unica catena da  $r$  a  $u$  è un **antenato di  $u$**  e  $u$  è un **discendente di  $r$** .

Un albero radicato può essere agevolmente rappresentato dalla seguente tabella che realizza la funzione padre.

<i>Figlio</i>	<i>Padre</i>
⋮	⋮
x	padre di x
⋮	⋮
⋮	⋮

Per definizione i figli non sono connessi fra di loro da alcuno spigolo ed esiste un solo spigolo che li congiunge alla radice.

Possiamo però far partire per ogni nodo figlio della radice vari rami per congiungerli a nuovi nodi che chiameremo i figli di questo nodo e che verrà chiamato a sua volta il loro padre. La terminologia e i vincoli iniziali vengono estesi ai nuovi figli e ai figli dei figli e così via ricorsivamente.

Quindi, per definizione di albero, ogni nodo ha un solo padre e un solo spigolo li congiunge, ma può avere a sua volta molti figli connessi ciascuno a lui con un solo ramo.

I nodi che non hanno figli vengono chiamati **foglie** e i nodi con almeno un figlio vengono chiamati **nodi interni**.

Un nodo  $A$  viene definito **progenitore** di un nodo  $B$  se esiste un nodo  $C$  figlio di  $A$  e padre di  $B$  o se esiste un nodo  $D$  figlio di  $A$  e progenitore di  $B$ . Si tratta di una definizione ricorsiva.

Poiché gli alberi sono strutture matematiche definite in modo ricorsivo, esprimere definizioni, proprietà o algoritmi sugli alberi in modo ricorsivo risulta spesso il modo più semplice e naturale.

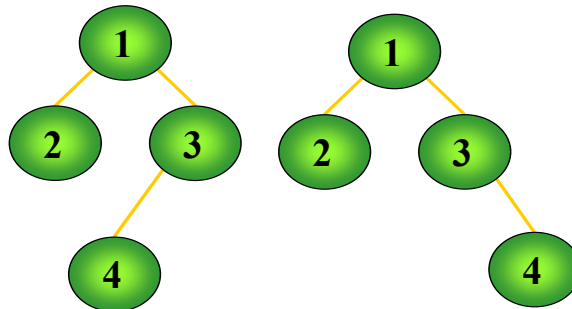
In senso più lato anche i padri vengono detti progenitori e i nodi di uno stesso progenitore si dicono **dominati** da quel progenitore.

Tutti i nodi di un albero sono dominati dalla radice.

### 1.5.2 ALBERI BINARI

Un albero binario è un albero radicato in cui ogni nodo interno ha al più due figli; ogni figlio è distinto come figlio sinistro oppure figlio destro.

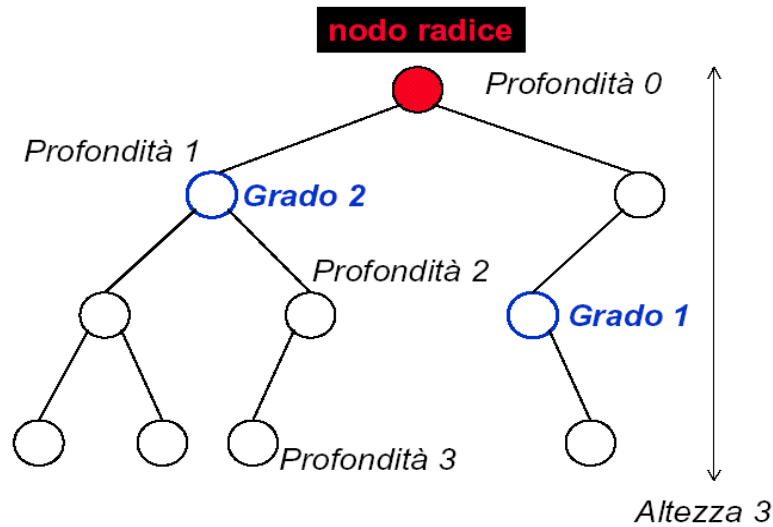
I seguenti due alberi sono coincidenti come alberi ordinati, ma distinti come alberi binari:



**Il livello** di un nodo è definito nel modo seguente: ogni nodo ha livello  $p+1$ , dove  $p$  è il livello del nodo padre con la convenzione che la radice ha livello zero.

**Il grado** di un nodo è il numero di sottoalberi di quel nodo.

**La profondità** di un albero è il massimo del livello dei suoi nodi. L'albero vuoto ha profondità zero. La lunghezza della catena più lunga, dalla radice ad una foglia, è l'**altezza** dell'albero.



Un albero binario con profondità  $k$  ha al minimo  $k+1$  nodi (albero degenere) e al più  $2^{k+1}-1$  nodi (albero pieno). Un albero degenere è di fatto una lista di  $k$  elementi mentre in un albero pieno, tutti i nodi a livello inferiore di  $k$  sono pieni per cui ha  $2^0=1$  nodo al livello zero,  $2^1$  nodi al primo livello,  $2^2$  nodi al secondo livello e così via fino a  $2^k$  nodi all'ultimo livello e quindi il numero totale di nodi è:

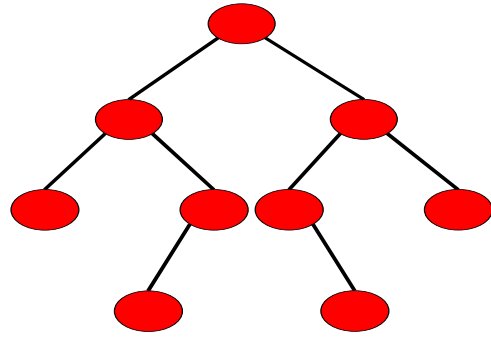
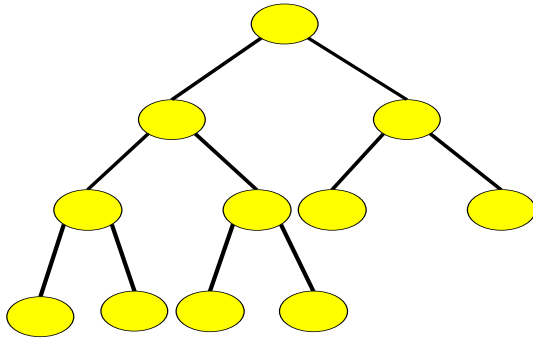
$$n = \sum_{j=0}^k 2^j = 2^{k+1} - 1$$

Un albero binario di profondità  $k$  è **completo** se i nodi dei livelli inferiori a  $k$  formano un albero pieno e per ogni nodo il sottoalbero di sinistra ha profondità non minore di quello di destra. In altre parole in un albero completo tutti i livelli tranne l'ultimo sono riempiti e l'ultimo livello è riempito da sinistra a destra per cui eventuali nodi mancanti sono collocati tutti nella parte destra.

Alberi completi (o “quasi” completi) contengono un gran numero di nodi con una bassa altezza. Viceversa, l'albero binario con  $n$  nodi che ha altezza massima è quello nel quale ogni nodo interno possiede un solo figlio. In questo caso l'altezza dell'albero è chiaramente  $h=n-1$ .

Un albero binario è **bilanciato** se per ogni nodo la profondità del sottoalbero di destra e la profondità del sottoalbero di sinistra differiscono al più di uno.

Per capire meglio quanto detto sopra facciamo due esempi:



a) Albero completo di profondità 3

b) Albero bilanciato di profondità 3

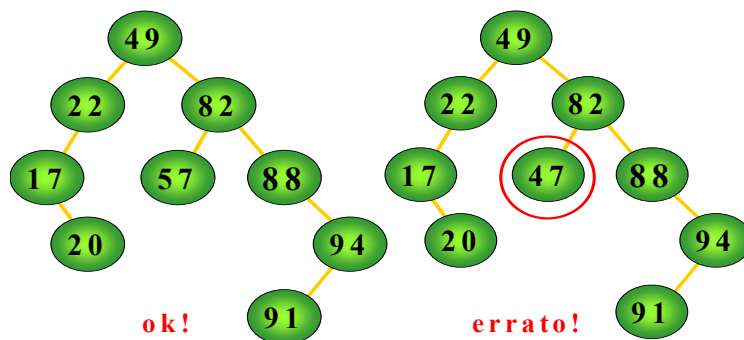
Un albero binario completo è bilanciato.

Quando un albero si riduce a una struttura lineare abbiamo massimo sbilanciamento.

### 1.5.2.1 ALBERI BINARI DI RICERCA

Un albero binario di ricerca è un albero binario tale che per ogni nodo p:

- viene associata una chiave  $key(p)$
- le chiavi del sottoalbero sinistro di p sono  $\leq key(p)$
- le chiavi del sottoalbero destro di p sono  $\geq key(p)$

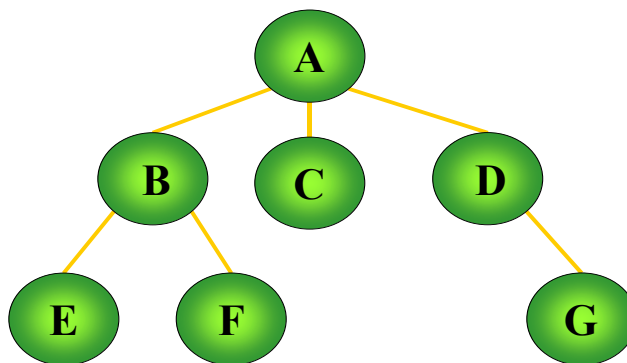


La verifica di appartenenza di un elemento all'insieme può essere effettuata efficientemente nel modo seguente (*ricerca binaria*):

- se l'albero è vuoto si ha un fallimento;
- altrimenti si confronta l'elemento da cercare con la radice dell'albero e
  - se gli elementi sono uguali ci si ferma con un successo;
  - se l'elemento da cercare è minore della radice, si prosegue la ricerca nel sottoalbero sinistro;
  - se l'elemento da cercare è maggiore della radice, si prosegue la ricerca nel sottoalbero destro.

### 1.5.3 ALBERI N-ARI

Arità = massimo numero di figli di qualche nodo

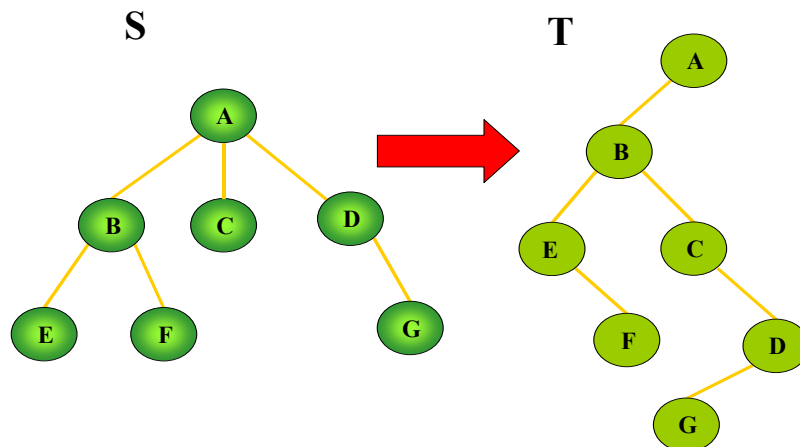


In questo caso l'albero è ternario.

### 1.5.3.1 TRASFORMAZIONE DA ALBERI N-ARI IN ALBERI BINARI

Qualsiasi albero può essere trasformato in un albero binario con la tecnica memorizzazione figlio-fratello, ossia:

- Primo figlio a sinistra
- Primo fratello a destra

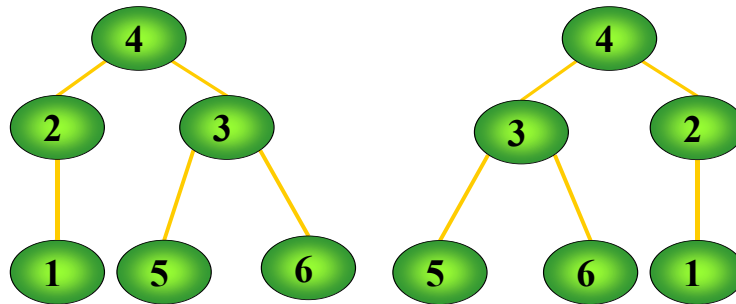


Cioè il numero di nodi di **S** e **T** coincidono, la radice di **S** coincide con la radice di **T** e ogni nodo  $n$  di **T** ha come radice del sottoalbero sinistro il primo figlio di  $n$  in **S** e come radice del sottoalbero destro il fratello successivo a destra di  $n$  in **S**.

Se  $n$  non ha i figli (fratelli a destra) in **S**, il sottoalbero sinistro (destro) di  $n$  in **T** è vuoto.

### 1.5.4 ALBERI ORDINATI

Un albero ordinato, detto anche piano, è un albero radicato in cui i figli di ogni vertice sono totalmente ordinati. I seguenti due alberi sono coincidenti come alberi radicati, ma distinti come alberi ordinati:

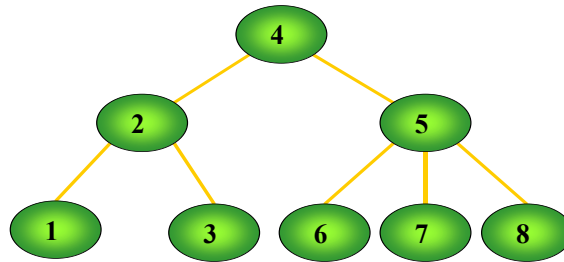


Una classica rappresentazione in memoria di un albero ordinato consiste nell'uso di tre vettori: se l'albero è formato da  $n$  nodi, rappresentati dai primi  $n$  interi, i tre vettori  $P$ ,  $F$ ,  $S$  hanno dimensione  $n$  e sono definiti nel modo seguente:

$$P[i] \equiv \begin{cases} j & \text{se } j \text{ é il padre di } i, \\ 0 & \text{se } i \text{ é la radice;} \end{cases}$$

$$F[i] \equiv \begin{cases} j & \text{se } j \text{ é il primo figlio di } i, \\ 0 & \text{se } i \text{ é una foglia;} \end{cases}$$

$$S[i] \equiv \begin{cases} j & \text{se } j \text{ é il fratello successivo di } i, \\ 0 & \text{se } i \text{ non possiede un fratello successivo;} \end{cases}$$



I tre vettori sono definiti da:

	<i>P</i>	<i>F</i>	<i>S</i>
1	2	0	3
2	4	1	5
3	2	0	0
4	0	2	0
5	4	6	0
6	5	0	7
7	5	0	8
8	5	0	0

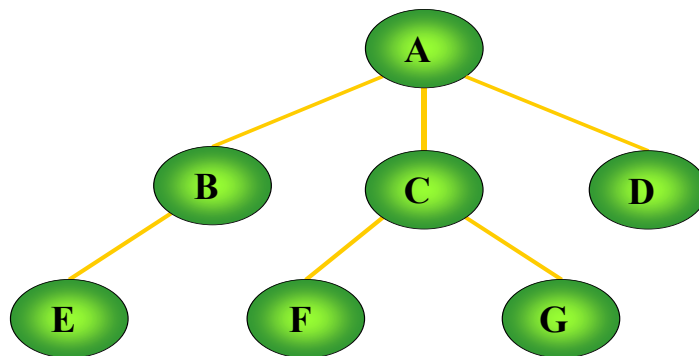


## 1.6 L'ORDINAMENTO

Per attraversamento o visita di un albero si intende l'ispezione dei nodi dell'albero in modo che tutti i nodi vengano ispezionati una e una sola volta. Quindi un attraversamento di un albero definisce un ordinamento totale, tra i nodi di un albero, in base alla loro posizione nell'albero e non in base alle loro etichette. Trattandosi di un ordinamento totale, ogni nodo ha un predecessore e un successore all'interno di un attraversamento.

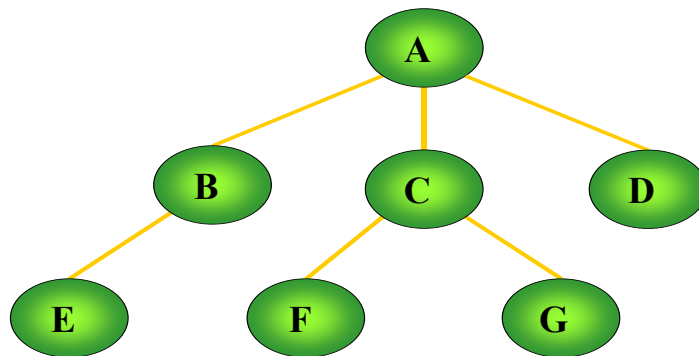
Esistono due modi/strategie/algoritmi fondamentali per visitare tutti i nodi di un albero partendo dalla radice, vengono chiamati **depth-first-search (DFS)** e **breath-first-search (BFS)**. I loro nomi derivano da quello che ciascun metodo decide di mettere al primo posto nella sequenza delle operazioni.

L'**algoritmo DFS** (visita in profondità) decide innanzi tutto di immergersi, secondo la logica alto-basso vista prima, scendendo i livelli fino a quando non raggiunge il fondo per poi risalire livello per livello e ricominciare con una nuova immersione. Inizia scegliendo un figlio a caso della radice, poi un figlio a caso del nodo trovato e così via fino a che non incontra una foglia. A questo punto risale al padre della foglia e cerca un altro figlio per ricominciare l'immersione. Se non è possibile immergersi in un nodo non ancora visitato risale di un livello e riparte con i restanti nodi figli. Quando, tornato alla radice, risulta che i figli di quest'ultima sono stati tutti già considerati, l'algoritmo termina avendo visitato tutti i nodi.



- **Visite in profondità/preordine** (anticipata se si visitano alberi binari): un nodo è ispezionato prima dei suoi discendenti. (Es.: A,B,E,C,F,G,D)
- **Visite in profondità/in ordine** (simmetrica se si visitano alberi binari): un nodo è ispezionato dopo il suo sottoalbero sinistro e prima del suo sottoalbero destro. (Es.: E,B,A,F,C,G,D)
- **Visite in profondità/postordine** (differita se si visitano alberi binari): un nodo è ispezionato dopo i suoi discendenti. (Es.: E,B,F,G,C,D,A)

**L'algoritmo BFS** (visita in ampiezza) parte dal nodo padre e successivamente decide di raccogliere, in un insieme, tutti nodi che può raggiungere con un solo spigolo. Man mano che visita i nodi di questo insieme, raccoglie in un nuovo insieme i nodi che può raggiungere con un solo spigolo partendo da uno dei nodi dell'insieme precedente. Continuando in questo modo enumera tutti i nodi dell'albero. L'algoritmo termina quando l'ultimo insieme costruito risulta vuoto.



**Visita in ampiezza:** A,B,C,D,E,F,G.



## *CAPITOLO 2*

# IL RISCHIO OPERATIVO

### 2.1 DEFINIZIONE DI RISCHIO

Gli eventi dannosi dipendono da numerose cause, sia naturali che artificiali e coinvolgono in maniera diversa persone, beni e infrastrutture.

La necessità per meglio definire questi eventi, ai fini della prevenzione e della pianificazione delle emergenze, ha portato alla loro schematizzazione per poter analizzare e confrontare fenomeni diversi per intensità ed effetti, sia temuti che avvenuti.

E' stato così introdotto nella letteratura scientifica il concetto di **rischio**, inteso come misura della potenzialità di un evento dannoso.

Con rischio si intende la probabilità che un fenomeno potenzialmente dannoso possa avvenire in un determinato luogo ed in un determinato tempo provocando un atteso valore di danno.

Poiché nella valutazione del rischio, fenomeni molto complessi si traducono in espressioni numeriche, l'analisi di rischio non può dare risultati certi e quindi applicabili, ad esempio a livello normativo; tuttavia è molto utile per valutare fenomeni e danni attesi, e quindi pianificare azioni preventive d'intervento.

Il rischio si può riassumere nella seguente espressione:

$$\mathbf{R = P * V}$$

Dove:



RISCHIO

grado di perdite in conseguenza di un fenomeno naturale o artificiale;



PERICOLOSITA'

probabilità che un fenomeno potenzialmente dannoso si verifichi in un dato tempo e in una data area;



VULNERABILITA'

attitudine a subire danni di un elemento o gruppo di elementi esposti al rischio derivante da un fenomeno di determinata pericolosità.

Perché vi sia rischio il prodotto  $P \times V$  deve essere diverso da zero, cioè entrambi i fattori devono essere presenti. Non vi sarà, ad esempio, rischio sismico, cioè  $R=0$  nel caso avvengano molti terremoti di forte intensità ( $P$  alta) in un'area desertica ( $V=0$ ) oppure, la vulnerabilità sia molto elevata, situazione tipica di una metropoli, quando questa si trova in un'area non sismica ( $P=0$ ).

## 2.2 RISCHIO SPECULATIVO E PURO

Il rischio in quanto conseguenza di fenomeni potenzialmente dannosi, viene diviso in due classi: il rischio speculativo (detto anche d'impresa) e il rischio puro (detto anche aleatorio).

- *Il rischio speculativo* è quello che corre un imprenditore nella gestione dei suoi affari, per il fatto che essi possono portare, come risultato, a guadagni ma purtroppo anche a perdite.
- *Il rischio puro* è un rischio che sorge all'improvviso e determina sempre delle perdite (incendio, furto, infortunio, ecc.).

Secondo questi elementi possiamo identificare delle categorie di massima per classificare i tre diversi tipi d'investitore.

Sono principalmente tre: conservativo, equilibrato e speculativo.

L'investitore conservativo punta sostanzialmente a mantenere intatto il valore del suo patrimonio, sottraendolo all'effetto erosivo dell'inflazione piuttosto che ad incrementarlo. In questo caso ovviamente il livello di rischio dell'investimento dovrà essere ridotto al minimo, anche se ciò si tradurrà inevitabilmente in rendimenti non esaltanti.

Qualche rischio in più sarà invece disposto a correrlo un investitore equilibrato mentre l'investitore di tipo speculativo andrà a caccia di altri rendimenti anche se questo potrebbe tradursi, per il suo capitale, in perdite più o meno rilevanti.

## **2.3 LE TIPOLOGIE DI RISCHIO**

Valutare il rischio non è facile e oltretutto, non esistono strumenti idonei a tale scopo. Tuttavia il problema può essere affrontato analizzando in dettaglio gli aspetti che determinano il rischio:

- *Rischio specifico*: è il rischio che grava su un'azienda in funzione dei suoi punti di forza e di debolezza. Tale fattore influenza i flussi di cassa attesi che varieranno in funzione della capacità dell'impresa di far fronte in modo adeguato a tale rischio.

Il rischio specifico è costituito a sua volta da:

- *Rischio finanziario*: correlato alla struttura finanziaria dell'impresa e alle scelte che incidono sulla stessa, rappresenta la potenziale variazione dei valori immobiliari in seguito a variazioni nelle variabili monetarie e finanziarie (tassi d'interesse, inflazione, liquidità).

- *Rischio operativo*: correlato alla posizione competitiva dell'impresa e alle variabili da cui si generano i risultati operativi attesi.
- *Rischio di business*: è il rischio del settore, correlato alle caratteristiche del business; coinvolge tutte le imprese facenti parte del settore. E' il fattore chiave per la definizione del costo del capitale, poiché influenza le aspettative di redditività degli investitori. Con un rischio di business elevato, gli investitori sono disposti ad investire solo in presenza di una redditività attesa elevata. Per le banche e per gli investitori finanziari, i rischi di business sono prevalentemente rischi di mercato e di credito:
  - *Rischio di mercato*: rischio di variazioni di valore dovute a movimenti avversi ai prezzi;
  - *Rischio di credito*: rischio d'insolvenza della controparte;
- *Rischio di sistema*: è il rischio legato alla parte normativa, amministrativa e ambientale.

## **2.4 IL RISCHIO OPERATIVO IN OTTICA BASILEA 2**

### **2.4.1 COS'E' IL COMITATO BASILEA 2?**

Nel 2006 entrerà in vigore un nuovo schema di adeguatezza patrimoniale<sup>5</sup> noto anche come **Nuovo Accordo di Basilea** o **Basilea 2**. Il nuovo schema è volto a conferire maggiore rilevanza alla gestione del rischio e a promuovere il costante potenziamento delle capacità di valutazione del rischio da parte delle banche.

La vigilanza bancaria internazionale è regolata da accordi stipulati nell'ambito di un Comitato le cui attività si svolgono con il sostegno della Banca per i Regolamenti Internazionali.

Lo scopo del Comitato può essere raggiunto attraverso una stretta correlazione dei requisiti patrimoniali delle banche, con le moderne prassi prevalenti in tema

---

<sup>5</sup> Adeguatezza patrimoniale: il patrimonio deve essere adeguato ai rischi assunti.

della gestione del rischio. Particolarmente rilevante a questo riguardo è l'introduzione del rischio operativo.

Il Nuovo Accordo si prefigge di adeguare le regole prudenziali al nuovo contesto caratterizzato da una maggior complessità dei rischi aziendali, più raffinate tecniche di valutazione dei rischi e presenza di nuovi sofisticati sistemi finanziari.

Basilea 2 si articola su tre pilastri:

### **1) I requisiti patrimoniali minimi**

Il primo pilastro è, in sostanza, un affinamento della misura prevista dall'accordo del 1988.

Fondamento dell'attuale Accordo è una definizione di coefficiente patrimoniale in cui il numeratore rappresenta l'ammontare di capitale a disposizione di una banca e il denominatore una misura dei rischi cui questa è esposta (e, come tale, coincide con la definizione di attività ponderate per il rischio). Il coefficiente patrimoniale che ne risulta non può essere inferiore all'8%.

Nella nuova formulazione dell'Accordo, le regole che definiscono il numeratore del rapporto patrimoniale – o, in altri termini, il patrimonio a fini di vigilanza – restano invariate. Ugualmente, non viene modificato il coefficiente minimo richiesto dell'8%. Di conseguenza, i cambiamenti intervengono in ciò che attiene alla definizione di attività ponderate per il rischio, ovvero nelle metodologie impiegate per misurare i rischi in cui incorrono le banche. I nuovi metodi per il calcolo delle attività ponderate sono volti a migliorare la valutazione della rischiosità da parte delle istituzioni bancarie e, pertanto, a rendere più significativi i coefficienti patrimoniali che da quella derivano.



## **2) Il controllo delle Banche Centrali**

Il secondo pilastro del Nuovo Accordo si basa su una serie di principi guida, improntati nella loro totalità alla duplice esigenza: che le banche valutino l'adeguatezza patrimoniale in rapporto ai loro rischi complessivi, e che le autorità di vigilanza verifichino tali valutazioni e assumano le opportune azioni correttive. Questi elementi sono ritenuti sempre più necessari ai fini, di un'efficace gestione delle organizzazioni bancarie e di una efficace vigilanza bancaria. Le valutazioni del rischio e dell'adeguatezza patrimoniale devono spingersi oltre la semplice verifica dell'osservanza da parte di una banca dei requisiti patrimoniali minimi. L'inclusione nel Nuovo Accordo di una componente relativa al controllo prudenziale apporta quindi notevoli vantaggi, in virtù del rilievo attribuito alla necessità che sia le banche sia gli organi di vigilanza dispongano di comprovate capacità di valutazione del rischio.

## **3) Disciplina del Mercato e Trasparenza**

Scopo del terzo pilastro è quello di integrare i requisiti patrimoniali minimi stabiliti nel primo pilastro e il processo di controllo prudenziale affrontato dal secondo. Il Comitato si è adoperato per incoraggiare la disciplina di mercato mediante l'elaborazione di una serie di obblighi di trasparenza che consentano agli operatori di valutare le informazioni cruciali sul profilo di rischio e sui livelli di capitalizzazione di una banca. Il Comitato reputa che il processo informativo assuma una particolare rilevanza con riferimento al Nuovo Accordo, laddove il ricorso a metodologie interne di valutazione conferirà alle banche una maggiore discrezionalità nel determinare il proprio fabbisogno di capitale. Spingendo in direzione di una più rigorosa disciplina di mercato, tramite il potenziamento delle segnalazioni, il terzo pilastro del nuovo schema patrimoniale potrà arrecare notevoli benefici a banche e autorità di vigilanza nella gestione del rischio e nel rafforzamento della stabilità.

Facciamo un esempio concreto: una banca finanzia le imprese *A* e *B* entrambe per 100 €.

- *Prima di Basilea 2:*

l'impresa *A*, con basso rischio di credito, corrisponde interessi al 4%; l'impresa *B*, a più alto rischio, interessi pari al 5%. L'utile della banca al netto di rettifiche pari a 2 €, sarà pari a 7 (4+5-2).

- *Dopo Basilea 2:*

la banca, potendo valutare meglio il rischio di credito, finanzia l'impresa *A* al 3% e l'impresa *B* al 6%. L'utile della banca in presenza di rettifiche pari al caso precedente, sarà sempre pari a 7 € (3+6-2), ma il costo dell'indebitamento per l'impresa più efficiente sarà diminuito da 4 € a 3 €.

## 2.4.2 ASPETTI PROBLEMATICI DI BASILEA 2

Sul documento originario di Basilea 2 sono state formulate alcune critiche che hanno portato a modifiche che, pur non cancellando i dubbi, dovrebbero attenuare le conseguenze negative attese dall'applicazione dell'accordo. Queste conseguenze negative sono tre:

### 1) **La difficoltà di raccogliere informazioni**

Il problema principale per le banche è quello di raccogliere le informazioni e i dati necessari per poter realizzare le metodologie più avanzate per la misurazione del rischio. Le piccole banche, che non dispongono in genere di forme evolute di *Risk Management*<sup>6</sup> potrebbero essere soggette a requisiti patrimoniali più stringenti. La discriminazione tra banche sarà ancor più rilevante in quanto il Nuovo Accordo rischia di aumentare l'onere patrimoniale delle singole banche. Questo rischio deriva dal fatto di aver introdotto un preciso requisito patrimoniale anche per il rischio operativo,

---

<sup>6</sup> Risk Management: strategie gestionali finalizzate al controllo di rischi aziendali.

imponendo un vincolo patrimoniale eccessivamente oneroso soprattutto per banche medio-piccole. Queste ultime si troveranno quindi nella situazione o di dover aumentare il costo del credito rischiando di perdere quote di mercato o di “prezzare” i propri strumenti in modo inadeguato, peggiorando in questo modo la propria situazione finanziaria e patrimoniale.

## **2) I rating interni**

Da un punto di vista teorico l’effetto di una più accurata valutazione del rischio da parte delle banche dovrebbe essere quello di ridurre il costo del credito per le imprese meno rischiose e di promuovere un rapporto fondato sulla conoscenza e sulla fiducia reciproca. In pratica però, vi è il rischio che i crediti concessi alle imprese medio-piccole (Pmi) siano, o continuino ad essere considerati più rischiosi: la metodologia di determinazione dei *rating interni*<sup>7</sup>, basata su procedure automatizzate (*scoring*), potrebbe rivelarsi poco adatta a cogliere le peculiarità delle Pmi, a valutarne cioè adeguatamente il merito di credito. Vi è quindi il timore che il metodo dei rating interni penalizzi il finanziamento delle Pmi, inducendo le banche a ridurre il credito ad esse destinato e ad aumentare al contempo i tassi d’interesse.

## **3) Il problema della prociclicità finanziaria**

Questo problema, già presente nell’originario Accordo del 1988, rappresenta una fonte di preoccupazione ai fini della stabilità finanziaria e macroeconomica. Quando il capitale o le riserve accumulati durante i periodi di espansione non sono sufficienti a coprire i rischi associati alle fasi di rallentamento congiunturale, le banche sono costrette a ridurre gli impieghi per assolvere ai requisiti patrimoniali. La regolamentazione patrimoniale può

---

<sup>7</sup> Rating significa alla lettera “valutazione” ; per le imprese rappresenta un voto oggettivo della capacità di credito, sulla solvibilità e solidità di bilancio. Alla base vale che “peggiore è il rating maggiore sarà la probabilità di perdita per la banca”, il che significa meno credito concesso alle imprese e costi del denaro più alti. I rating interni vengono calcolati dalle banche ed il loro costo è gratuito per coloro che richiedono finanziamenti.

influire sulla prociclicità finanziaria. Ad esempio nei periodi di rallentamento economico, in cui cresce la rischiosità dell'attivo, le banche sono indotte ad accantonare maggiore capitale. Per una banca il cui grado di copertura sia al limite dell'8%, questo deve necessariamente avvenire a scapito di nuovi prestiti o del rinnovo di prestiti già esistenti.

## 2.5 OPERATIONAL RISK

Focalizziamo ora l'attenzione sull'introduzione del rischio operativo definito come “il rischio di perdite derivanti da processi, comportamenti del personale o sistemi interni inadeguati o non andati a buon fine, oppure derivanti da eventi esterni” .

Tre sono i fattori da tenere presente nell'analisi di questa tipologia di rischio:

- la sicurezza;
- il disegno, la realizzazione e la manutenzione dei processi;
- il cattivo uso dei prodotti e servizi da parte della clientela.

Per quanto concerne la sicurezza, l'accento va posto sui controlli di accesso dall'esterno ai sistemi bancari (ad esempio, il rischio di essere soggetti ad attacchi da parte degli hacker) e sulle possibilità di subire danni da parte del personale della stessa banca, sia sotto il profilo dell'errore umano che della frode vera e propria. Per il disegno, la realizzazione e la manutenzione dei processi, il riferimento è per tutte le problematiche che possono derivare da deficienze nel funzionamento del sistema stesso, legate, ad esempio, a rallentamenti, interruzioni oppure all'obsolescenza della tecnologia adottata. Infine, vi è il rischio del cattivo uso dei prodotti e servizi da parte della clientela, che può essere sia involontario che intenzionale (per quest'ultimo, ad esempio, il riciclaggio di denaro).

Il Comitato Basilea 2 considera tale rischio un fattore particolarmente critico per le banche e reputa pertanto necessario che queste si cautelino da possibili perdite con una opportuna dotazione di capitale. Inoltre viene concessa alle

banche una eccezionale flessibilità nell’elaborazione di una metodologia di calcolo del patrimonio a fronte del rischio operativo che sia ritenuta coerente con il profilo della loro operatività e dei connessi rischi.

La principale innovazione del Nuovo Accordo proposto consiste nell’introduzione di tre distinte opzioni di calcolo per il rischio operativo. Questi metodi consentono a banche e autorità di vigilanza di scegliere, quello o quelli, ritenuti più appropriati allo stadio di sviluppo dell’operatività bancaria e dell’infrastruttura di mercato. La tabella seguente riporta i tre metodi disponibili.

<b>Rischi Operativi</b>
1)Metodo dell’indicatore semplice (“basic Indicator Approach”)
2)Metodo Standard
3)Metodi avanzati di misurazione (“Advanced Measurement Approaches” – AMA)

## **2.5.1 METODI DI CALCOLO PER I RISCHI OPERATIVI**

### **2.5.1.1 METODO DELL’INDICATORE SEMPLICE (BIA)**

Tra le tre metodologie di calcolo del requisito patrimoniale, a fronte del rischio operativo, proposte nel nuovo Accordo, questa risulta sicuramente la più semplice.

Questo approccio correla il rischio operativo a una dotazione di capitale espressa come percentuale fissa di una specifica misura del rischio stesso; tale misura corrisponde alla media del reddito lordo annuo della banca calcolata sui tre esercizi precedenti.

Il coefficiente patrimoniale può essere espresso come segue:

$$C_{BIA} = GI \cdot \beta$$

dove:

- $C_{BIA}$  = coefficiente patrimoniale definito nel “basic indicator approach”.
- $GI^8$  = reddito lordo medio annuo riferito ai tre esercizi precedenti; tale reddito è definito come (reddito netto da interessi + reddito netto non da interessi): in questo modo tale reddito dovrebbe essere al lordo di ogni accantonamento, esclude i profitti o le perdite realizzate sulla vendita di titoli del “banking book”, ed infine esclude le partite straordinarie, gli errori di omissioni, nonché i provenienti derivanti da assicurazioni.
- $\beta = 15\%$  (percentuale stabilita dal Comitato), rappresenta il reddito lordo medio annuo.

Per ciò che concerne il calcolo del patrimonio di vigilanza, il Nuovo Accordo non fissa criteri specifici per l’impiego del metodo dell’indicatore semplice. Nonostante ciò, le banche che ricorrono a tale approccio sono invitate a conformarsi alle linee guida in materia di prassi corrette per la gestione e il controllo del rischio operativo, pubblicate dal Comitato nel febbraio 2003.

### **2.5.1.2 METODO STANDARD (MS)**

Anche il metodo standard si avvale del reddito lordo come indicatore di massima per la scala dimensionale dell’attività operativa della banca e, quindi, della probabile entità della connessa esposizione al rischio operativo per una data linea di attività. Peraltro, anziché calcolare il patrimonio a livello di intera azienda, come previsto dall’approccio dell’indicatore semplice, le banche devono procedere in questo caso al computo del requisito di capitale per ciascuna linea operativa. Questo viene determinato moltiplicando il reddito lordo per specifici fattori prudenziali decisi dal Comitato. Il coefficiente patrimoniale complessivo, a fronte del rischio operativo, per una organizzazione bancaria sarà dato dalla

---

<sup>8</sup> GI sta per Gross Income cioè margine d’intermediazione.

sommatoria dei singoli coefficienti patrimoniali riferiti a ogni singola linea operativa.

Basilea 2 suddivide l'attività bancaria in otto linee di business:

- corporate finance;
- negoziazioni e vendite di strumenti finanziari;
- servizi di pagamento;
- commercial banking;
- gestioni finanziarie;
- rateil banking;
- asset management;
- negoziazione di titoli al dettaglio;

e il coefficiente patrimoniale può essere espresso come:

$$C_{STA} = \sum_{i=1}^8 (GI_i \cdot \beta_i)$$

dove:

- $C_{STA}$  = coefficiente patrimoniale definito nel “standard approach”.
- $GI$  = reddito lordo medio annuo riferito ai tre esercizi precedenti per ciascuna delle otto linee operative;
- $\beta$  = percentuale fissa, stabilita dal Comitato, che rapporta il livello di capitale richiesto a quello del reddito lordo per ciascuna delle otto linee; per le prime tre linee di business, visto il loro maggior contenuto di rischio, il coefficiente del 15% viene alzato al 18%, simmetricamente il coefficiente scende al 12% per le ultime tre, ritenute meno rischiose.

E' da segnalare altresì l'opzione alternativa per lo *Standard Approach*, applicabile in due delle otto linee operative (*retail banking* e *commercial banking*), che prevede la sostituzione dell'indicatore costituito dal reddito lordo determinato a livello di *business line* con il totale lordo annuo dei prestiti e delle anticipazioni, rispettivamente verso la clientela *retail* e verso le imprese,

ponderato per un coefficiente  $m$  pari al 3,5%. Quindi il coefficiente patrimoniale per il rischio operativo nell’approccio Standard Alternativo per il “retail banking” (con la stessa formula base per il “commercial banking”) può essere espresso come segue:

$$C_{RB} = \beta_{RB} \cdot m \cdot LA_{RB}$$

dove:

- $C_{RB}$  = coefficiente patrimoniale per il “retail banking”;
- $\beta_{RB}$  = è il fattore  $\beta$  per il “retail banking”;
- $LA_{RB}$  = è la media degli ultimi tre esercizi del totale dei prestiti e delle anticipazioni “retail” in essere (non ponderate per il rischio e al lordo degli accantonamenti);
- $m = 3,5\%$ .

### **2.5.1.3 METODI AVANZATI DI MISURAZIONE (AMA)**

Si tratta di un modello interno della banca per la misurazione del rischio operativo, soggetto all’approvazione dell’organo di vigilanza.

Il Comitato ha tenuto presente il rapido sviluppo delle tecniche interne di valutazione presso le organizzazioni bancarie, cercando di fornire loro incentivi a migliorare in futuro tali dispositivi nonché, più in generale, l’intera gestione del rischio operativo. Ciò assume una particolare rilevanza nel caso degli approcci avanzati di misurazione (“Advanced Measurement Approaches” – AMA) .

Un AMA è basato essenzialmente su dati interni di perdita, è flessibile ed esauriente nella misurazione dei rischi operativi ed è in grado di cogliere le esposizioni più significative. A questo proposito il Comitato Basilea ha definito rigorosi standard di natura quali/quantitativa che dovranno essere rispettati. Le banche dovranno rilevare sistematicamente i dati più importanti legati a queste fattispecie di rischi, inclusi i dati di perdita e, da un punto di vista qualitativo,



creare incentivi per migliorare la gestione dei rischi operativi all'interno dell'impresa nel suo complesso.

Requisiti addizionali sono poi richiesti per l'applicazione di un AMA, in particolare la programmazione e l'attuazione di metodi di valutazione e strategie, nonché controlli e azioni correttive. E' necessario il rispetto di procedure rigorose, che passino per un sistema capillare di rilevazione dei dati di perdita, su base continuativa e per un periodo di osservazione minimo di 3-5 anni, al fine di giungere a stime empiriche della rischiosità. Sono perciò richieste adeguate competenze e risorse nella classificazione dei dati e nell'attività di allocazione delle perdite e degli eventi a specifiche *business line*. Potranno essere utilizzati anche dati esterni, provenienti da iniziative interbancarie organizzate su base consortile e si dovranno definire, in tali casi, procedure di raccordo tra le informazioni esterne e quelle disponibili internamente.

Per la determinazione del Capitale a Rischio il Comitato di Basilea ha individuato tre approcci alternativi:

- 1) **Internal Measurement Approach:** Le attività della banca sono suddivise in linee di business (come nell'approccio precedente), ma in più vengono individuati dei "tipi" di rischio operativo. Per ciascuna combinazione linea di business / tipo di rischio, viene calcolata una perdita attesa, che deve essere moltiplicata per uno specifico fattore moltiplicativo Gamma  $\gamma$ , al fine di ottenere il requisito sul rischio operativo ad essa relativo. La somma di questi requisiti specifici costituisce il requisito complessivo. Per il calcolo delle perdite attese per ciascuna combinazione *area operativa/tipologia di rischio* le istituzioni potranno fornire le stime di tre elementi:
  - **EI** (*Exposure indicator*): indicatore di esposizione ai rischi operativi dell'area.
  - **PE** (*Probability of loss event*): probabilità che si verifichi l'evento dannoso. La probabilità di un evento è data dal rapporto tra eventi favorevoli ed eventi possibili.

- **LGE** (*Loss given that event*): entità della perdita nel caso si verifichi l'evento dannoso.

Una volta effettuate le stime si calcola la **perdita attesa** (*Expected Loss*) è:

$$EL = EI \cdot PE \cdot LGE$$

Le autorità di vigilanza indicano un fattore Gamma  $\gamma$  per ciascuna combinazione *area operativa/tipologia di rischio* che traduce la perdita attesa in un requisito patrimoniale ed è fissato dalle autorità di vigilanza.

Requisito patrimoniale =  $\gamma$  \* EL (per ciascuna combinazione)

Requisito patrimoniale totale = sommatoria di tutti i requisiti patrimoniali.

- 2) **Loss Distribution Approach**: si tratta di un modello che stima per ogni area la distribuzione di perdite operative relative ad un certo orizzonte temporale. Per ciascuna combinazione *area operativa/tipologia di rischio*, viene effettuata una stima:

- della distribuzione di probabilità dell'impatto economico;
- della distribuzione di probabilità relativa alla frequenza dell'evento su un dato orizzonte temporale.

Dalle due distribuzioni precedenti viene determinata la distribuzione di probabilità delle perdite operative e il VaR di ogni linea operativa. Il requisito patrimoniale viene soddisfatto mediante la somma dei VaR elementari. Il VaR (value at risk) è la stima della massima perdita potenziale attesa in un arco temporale definito, con un certo grado di probabilità, a seguito del verificarsi di condizioni sfavorevoli; esso è in grado di quantificare la perdita con riferimento a ciascun fattore di rischio. In sostanza le banche stimano per ogni combinazione *linea operativa/tipologia di rischio* la distribuzione di probabilità delle perdite operative su un certo orizzonte temporale futuro.

- 3) **Scorecard Approach**: con questo approccio l'istituzione dovrebbe tradurre i giudizi qualitativi risultanti da un processo di valutazione (*scoring*) in stime quantitative previsionali, basate su indicatori di rischio. Gli *scorecards* sono sistemi che valutano variazioni della qualità del sistema di controllo interno della banca e del profilo di rischio sottostante. Nel Nuovo Accordo di Basilea gli *scorecards* devono avere solide basi quantitative e l'ammontare complessivo del capitale a copertura del rischio operativo deve essere comunque basato su una rigorosa analisi dei dati interni ed esterni delle perdite operative. Attraverso le risultanze degli *scorecards* è possibile calibrare il capitale complessivo sulle diverse linee operative.

La differenza tra i diversi approcci proposti per la misurazione dei rischi operativi non sta solo nel diverso grado di sofisticazione e di requisiti organizzativi delle strutture e processi aziendali, ma sta nel possibile **risparmio di capitale**.

La Banca d'Italia incoraggia le Banche, e in particolare i Gruppi bancari, all'adozione dei Metodi di tipo Avanzato, per il calcolo del capitale assorbito a fronte dei rischi operativi.

## *CAPITOLO 3*

# ANALISI TRAMITE ALBERO DI GUASTO

### **3.1 INTRODUZIONE**

Questa tesi è relativa ad un lavoro di stage, presso la società Engineering Ingegneria Informatica S.p.A., nell'ambito di un progetto riguardante la costruzione di un algoritmo che calcoli l'ordinamento di un albero di guasto.

Fondamentale per la realizzazione di questo progetto è stato imparare ad usare Mathcad, un software utile per effettuare calcoli particolarmente complessi e per rappresentare funzioni e grafici; questo software si distingue per la semplicità con cui si possono inserire i dati nel foglio di lavoro, grazie anche ad un'interfaccia rapida ed intuitiva. Il foglio Mathcad si presenta come una pagina bianca, in ogni punto della quale è possibile svolgere un calcolo, tracciare un grafico, definire una funzione, scrivere un testo...

Per quanto riguarda la programmazione è possibile costruire programmi di una certa complessità sfruttando un'ampia libreria di comandi.

In questo capitolo verrà specificato cosa si intende per Fault Tree Analysis e nel capitolo successivo verrà analizzato, in tutte le sue parti, l'algoritmo creato in Mathcad.

### 3.2 ENGINEERING INGEGNERIA INFORMATICA SPA

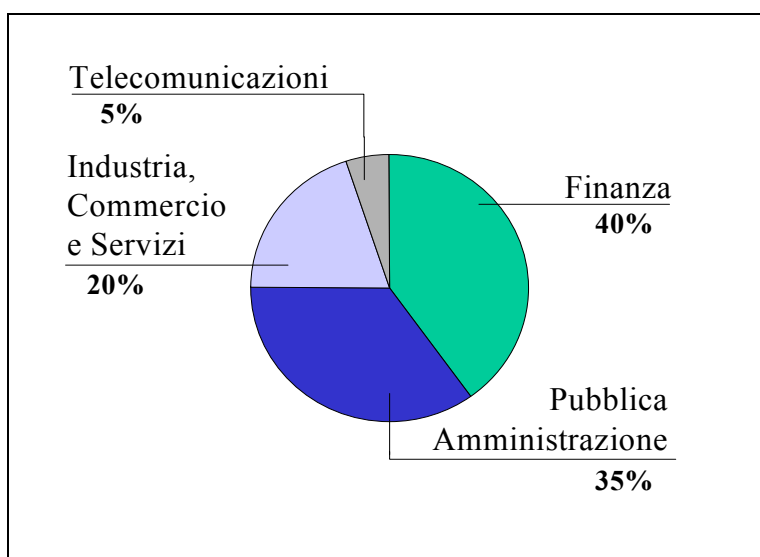
La società nasce nel 1980 con la denominazione Cerved Engineering S.p.A. a seguito del progetto per l'automazione delle Camere di Commercio italiane. Nel 1984 il management rileva interamente la quota di Cerved S.p.a. e dal 1985 la denominazione della società viene mutata in Engineering Informatica S.p.A.

Oggi, il Gruppo Engineering è composto da tredici società attive nel settore dell'Information Technology, in particolare, nell'attività di *business* e *system integration* e, più di recente, nell'erogazione di servizi di *outsourcing*.

Il Gruppo realizza grandi progetti informatici mediante l'integrazione di sistemi, reti, prodotti e tecnologie, nonché la fornitura di consulenza organizzativa, di revisione dei processi aziendali (*Business Process Reengineering*) e di servizi specialistici.

Le attività del Gruppo Engineering sono focalizzate su quattro segmenti del mercato:

- Finanza;
- Pubblica Amministrazione;
- Industria, Commercio e Servizi;
- Telecomunicazioni;



Le attività principali sono:

- **consulenza:** consiste nel verificare la rispondenza dei processi e delle infrastrutture tecnologiche aziendali del cliente alle necessità operative attuali ed emergenti, proponendo gli interventi più opportuni a livello organizzativo, informatico e di architettura dei sistemi;
- **studio di fattibilità:** consiste nell'analisi delle strategie del cliente e nell'identificazione delle possibili implementazioni in termini di assetti organizzativi, piattaforme e architetture tecnologiche, infrastrutture di rete e disponibilità di package. Per ogni ipotesi vengono effettuate analisi costi/benefici e analisi del rischio;
- **progettazione e realizzazione:** consiste nella fornitura di soluzioni software "chiavi in mano" comprendenti una o più delle seguenti attività: sviluppo di componenti software specifici; adattamento di prodotti software di mercato alle esigenze del cliente; adeguamento di componenti software preesistenti; dimensionamento delle apparecchiature hardware e delle reti; formazione degli utenti del nuovo sistema.

Le soluzioni Engineering sono quindi realizzate in relazione alle specifiche esigenze del cliente. Quando tali applicazioni vengono ritenute d'interesse per il mercato, la società procede alla loro standardizzazione per una più ampia commercializzazione.

### **3.3 IL RISK MANAGEMENT**

Il mondo industriale di oggi, per quanto sfaccettato e variegato, è contraddistinto da un'identica primaria esigenza: la protezione industriale, ovvero la protezione di tutti quegli elementi, materiali o immateriali, che sono alla base dei processi di creazione di valore per l'azienda stessa: le persone, i beni aziendali, l'ambiente, etc. In questo nuovo contesto di riferimento, l'azienda si deve porre due obiettivi fondamentali:

- la salvaguardia nel tempo del proprio patrimonio;
- la garanzia della continuità operativa, cioè la non-interruzione dei processi portatrici di valore, e quindi di ricchezza, all'azienda stessa.

Lo strumento, nato alcuni anni or sono ed oggi disponibile per raggiungere i due suddetti obiettivi, è il cosiddetto "RISK MANAGEMENT": questo termine indica la funzione aziendale che ha il compito di identificare, valutare, gestire e sottoporre a controllo economico i RISCHI PURI d'impresa, cioè quei rischi collegati ad eventi in grado di ridurre il valore aziendale, ovvero di dare luogo a perdite.

### **3.4 ANALISI DEI RISCHI AZIENDALI**

L'analisi dei rischi aziendali consiste in un lavoro di raccolta ed elaborazione di informazioni, teso a migliorare la conoscenza dei rischi ed aumentare la precisione degli interventi. Ci sono varie tipologie di analisi, Preliminary Hazard Analysis (PHA), Failure Models & Effect Analysis (FMEA), Hazard and Operability Studies (HAZOP) e Fault Tree Analysis di cui ci occuperemo più approfonditamente.

### 3.4.1 FAULT TREE ANALYSIS

L'analisi degli alberi di guasto o FTA (*Fault Tree Analysis*) è stata introdotta per la prima volta nel 1962 nei Bell Telephone Laboratories, in relazione allo studio della sicurezza di sistemi di controllo missilistico; negli anni successivi questo metodo è stato utilizzato sempre più diffusamente nell'industria aerospaziale e nucleare e, in generale, per lo studio di sistemi complessi di grosse dimensioni.

Essa è particolarmente adatta per l'analisi di sistemi altamente ridondanti, ossia per quei progetti in cui più possibili effetti di guasto scaturiscono da uno stesso gruppo di cause.

Per sistemi particolarmente vulnerabili a singoli guasti che possono provocare un incidente è invece consigliabile utilizzare tecniche di tipo diverso, come la FMEA<sup>9</sup> o la HAZOP<sup>10</sup>.

L'uso della FTA richiede una conoscenza dettagliata del funzionamento del sistema, dei modi di guasto dei suoi componenti e dei relativi effetti; un analista esperto e qualificato può sviluppare un albero di guasto ma è comunque necessario che il modello venga riesaminato da operatori o altro personale che abbiano esperienza operativa con il sistema ed i componenti in esso presenti.

Il tempo ed il costo di una FTA dipendono dalla complessità del prodotto in esame e dal livello di risoluzione dell'analisi: un sistema relativo ad un processo semplice può essere risolto da un analista in un giorno, mentre un sistema complesso, che richiede lo sviluppo di più alberi di guasto, può impegnare un gruppo di esperti anche per mesi.

Questo metodo può essere applicato sia durante la fase di progettazione, sia in fase di verifica di un prodotto già esistente allo scopo di migliorare la sicurezza

---

<sup>9</sup> FMEA è uno strumento di analisi dei processi ad alto rischio, serve per identificare i possibili eventi avversi ed i loro effetti.

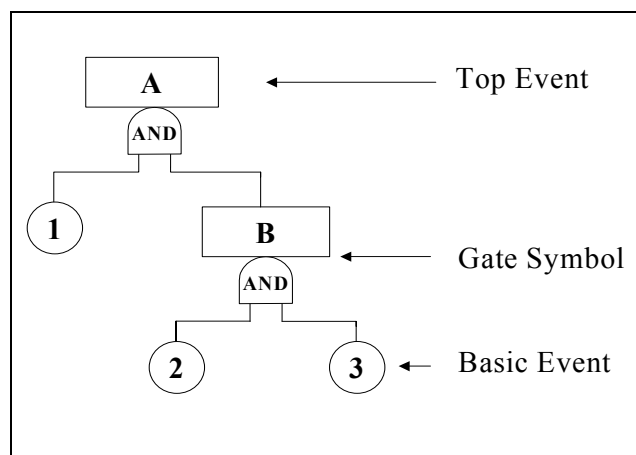
<sup>10</sup> HAZOP è una tecnica che si basa su un esame sistematico delle condizioni operative di funzionamento allo scopo di identificare gli interventi di modifiche strutturali e operative per prevenire e controllare le conseguenze di eventuali malfunzionamenti.



del sistema, dal momento che permette di individuare importanti caratteristiche come i punti deboli del sistema, false ridondanze o gli effetti di un dato componente sull'affidabilità complessiva.

La tecnica degli alberi di guasto richiede la decomposizione del sistema in un diagramma logico, detto *albero di guasto*, in cui certi eventi primari conducono ad uno specifico evento che rappresenta l'avaria totale del sistema, detto *Top Event* poiché si trova sulla sommità dell'albero di guasto.

Facciamo un esempio:



Iniziando dal *Top Event*, l'albero di guasto è costruito ramificandosi verso livelli più bassi costituiti da eventi intermedi che potrebbero determinare il *Top Event*: usando i comuni operatori logici AND, OR, NOT si ricostruisce la sequenza degli eventi fino a raggiungere gli eventi di base, di cui sono note le probabilità di accadimento.

Per evento di base si intende un qualunque guasto, o un qualunque errore umano, che porti alla rottura o al malfunzionamento di un componente o di un sottosistema dell'impianto in esame.

Durante la definizione dell'albero è inoltre importante considerare la presenza di eventuali *cause comuni di guasto*; se il malfunzionamento o la rottura di un componente del sistema considerato provoca la perdita di più funzioni del sistema stesso, i due o più effetti provocati non sono indipendenti tra loro e questo fatto

può portare ad una notevole variazione dell'espressione della possibilità di guasto totale, ovvero del *Top Event*.

Quando si sviluppa l'analisi FTA, gli insuccessi del componente (inputs) sono classificati in tre categorie: insuccesso primario, insuccesso secondario e insuccesso di comando.

- *Guasto primario* : l'insuccesso primario è dovuta alla costruzione o alle caratteristiche materiali del componente stesso.
- *Guasto secondario*: insuccesso del componente è causato da influenze esterne inaccettabili, come per esempio condizioni ambientali, condizioni di applicazione o l'influenza di altre componenti del sistema.
- *Guasto di comando*: è causato da errori di natura umana (operativi) o per uso scorretto.

Per rappresentare la sequenza di eventi che devono verificarsi per arrivare alla manifestazione del *Top Event*, sono presenti nell'albero, a vari livelli, dei nodi costituiti da *gate* che rappresentano graficamente le relazioni che intercorrono tra i rami che convergono in ciascun nodo.











I *gate* maggiormente usati sono:

- **OR**: l'evento di *output* si verifica se si verifica almeno uno degli *input*;
- **AND**: l'evento di *output* si verifica solo se si verificano tutti gli *input*;
- **NOT**: l'evento di *output* è la negazione dell'evento di *input*.
- **XOR** (OR esclusivo) l'evento di *output* si verifica se e solo se uno e non più di un evento di *input* è vero mentre risulta falso se lo è più di uno o nessuno.

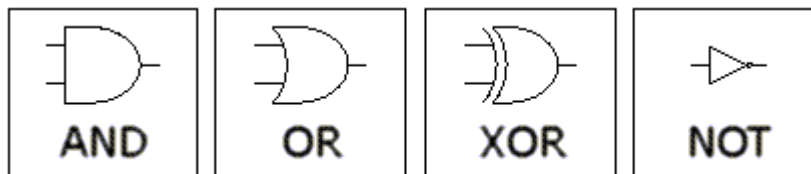
I suddetti operatori possono essere tra di loro combinati per formare equazioni logiche più complesse. Per esempio lo stesso operatore **XOR** è in realtà ottenibile a partire dagli altri tre, osservando che l'uscita è vera se A è falso e B vero, oppure se A è vero e B falso. Prendendo questa stessa frase, sostituendo le congiunzioni "e" e "oppure" con i corrispettivi operatori e l'asserzione falso con "non" vero, si ottiene l'equazione logica dello XOR.

$$A \text{ XOR } B = ((\text{NOT } A) \text{ AND } B) \text{ OR } (A \text{ AND } (\text{NOT } B))$$

Ciascun operatore ha anche un suo corrispondente simbolo **grafico**, la cui forma ne identifica il tipo.

Evento che rappresenta una gate		Blocco Gerarchico	
Evento di base		Evento di base negato	
Gate AND		Gate NAND	
Gate OR		Gate NOR	
Gate NOT		Gate EOR	

Il simbolo è anche dotato di piccoli segmenti sporgenti che rappresentano ingressi ed uscite, così da renderne semplice la combinazione in equazioni complesse.



### 3.4.2 L'ALGEBRA BOOLEANA

Il metodo sicuramente migliore per la risoluzione di un albero di guasto è quello che utilizza la sua espressione booleana<sup>11</sup> per definire la *funzione di guasto* caratteristica dell'albero, il cui valore rappresenterà lo stato del sistema in esame.

La possibilità di esprimere la funzione di guasto mediante un'espressione booleana è dovuta al fatto che si prendono in considerazione unicamente sistemi binari, in cui è possibile rappresentare lo stato di ogni componente, o di ogni evento di base, mediante delle variabili logiche binarie, dalle quali dipenderà la funzione di guasto stessa.

Si indica con  $X_S$  lo stato del sistema e con  $x_i$  quello del suo singolo componente:  $X_S$  e  $x_i$  sono delle variabili logiche che assumeranno valore 0 nel caso di corretto funzionamento del componente del sistema e valore 1 in caso contrario.

Mediante l'espressione booleana si arriva quindi a determinare una relazione del seguente tipo:

$$X_S = \Phi(x_i, \dots, x_n)$$

dove  $\Phi$  è la funzione di guasto ed  $n$  il numero degli eventi di base presenti nell'albero.

Per ricavare l'espressione della funzione di guasto è necessario conoscere il modo di operare dei singoli *gate* che possono essere presenti nell'albero di guasto; per i *gate* introdotti precedentemente valgono le seguenti proprietà:

- Proprietà di identità:  $x_i + 0 = x_i$  e  $x_i \cdot 1 = x_i$
- Proprietà di assorbimento:  $x_i + 1 = 1$  e  $x_i \cdot 0 = 0$
- Proprietà di idempotenza:  $x_i + x_i = x_i$  e  $x_i \cdot x_i = x_i$
- Proprietà dell'inverso:  $x_i + \bar{x}_i = 1$  e  $x_i \cdot \bar{x}_i = 0$
- Proprietà commutativa:  $x_i + x_j = x_j + x_i$  e  $x_i \cdot x_j = x_j \cdot x_i$

---

<sup>11</sup> 'algebra booleana' prende in nome da George Boole (England 1815, Ireland 1864), logico e matematico inglese, creò lo strumento concettuale che sta alla base del funzionamento del calcolatore.

- Proprietà associativa:  $x_i + (x_j + x_z) = (x_i + x_j) + x_z$  e  $x_i \cdot (x_j \cdot x_z) = (x_i \cdot x_j) \cdot x_z$
- Proprietà distributiva:  $x_i \cdot (x_j + x_z) = (x_i \cdot x_j) + (x_i \cdot x_z)$  e  $x_i + (x_j \cdot x_z) = (x_i + x_j) \cdot (x_i + x_z)$

Per effettuare i calcoli si possono quindi utilizzare gli operatori algebrici che, d'altra parte, vengono ricavati da quelli booleani mediante le seguenti relazioni:

$$\text{AND } x_i \cup x_j = x_i \cdot x_j$$

$$\text{OR } x_i \cap x_j = x_i + x_j - x_i \cdot x_j$$

$$\text{NOT } x_i = 1 - x_i = \bar{x}_i$$

La funzione di guasto, dopo gli opportuni calcoli e le opportune semplificazioni, viene espressa mediante una somma di prodotti.

In questa espressione, la variabile  $X_S$  assumerà valore 1 (che, ricordiamo, implica il verificarsi del *Top Event*) qualora anche uno solo degli addendi di tale somma assuma valore 1, come anche nel caso in cui tutti i fattori di un prodotto assumano valore 1 contemporaneamente.

Quindi si può affermare che ogni addendo della *somma di prodotti* rappresenta un *Cut-Set* (*stato del sistema tale che il Top Event si verifica*) dell'albero di guasto; se inoltre gli addendi sono tutti indipendenti tra loro, ovvero se nessuno di essi ne implica un altro, si può affermare che tali addendi rappresentano i cosiddetti *Minimal Cut-Set* dell'albero di guasto, ossia quel sottoinsieme di *Cut-Set* che non contiene nessun altro *Cut-Set*.

Intervenendo quindi su un qualsiasi componente facente capo ad un *Minimal Cut-Set* si può di fatto migliorare l'affidabilità del sistema in esame.

### **3.4.3 LE FASI DI UN'ANALISI FAULT TREE**

Lo sviluppo di una FTA può avvenire in quattro fasi:

- Identificazione del Top Evet
- Definizione del sistema
- Costruzione degli alberi di guasto
- Valutazione/analisi degli alberi di guasto

Che saranno illustrate successivamente.

#### **3.4.3.1 IDENTIFICAZIONE DEL TOP EVENT**

La prima fase consiste nell'identificare i Top Event che si vogliono analizzare a livello di sistema. Gli eventi di guasto per un sistema possono essere molteplici e con obiettivi di studio diversi: problemi di sicurezza verso l'esterno, di integrità, di disponibilità oppure di efficienza o di miglioramento delle politiche di manutenzione... e possono essere definiti a priori, in modo intuitivo o basandosi su risultati di altre analisi.

E' importante evidenziare che:

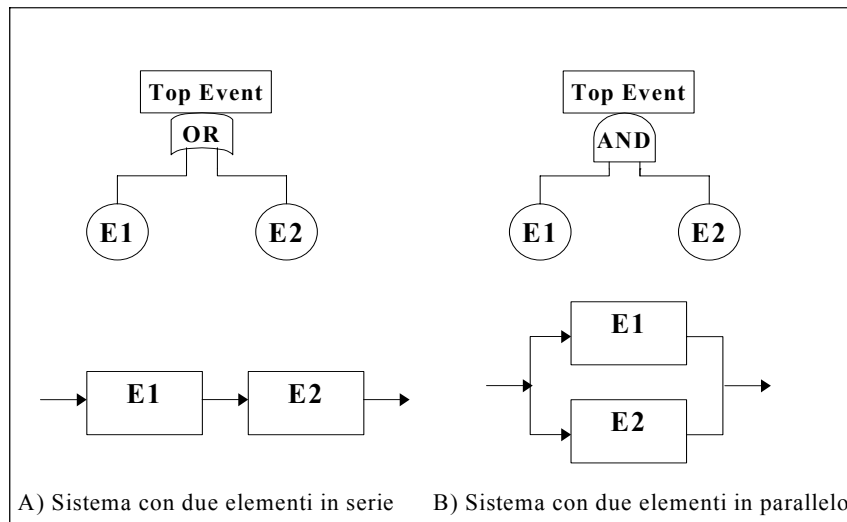
- Per ogni T.E. si elabora un albero, quindi il numero di eventi deve essere contenuto;
- Il singolo T.E. non va definito troppo genericamente perché si incorre in una lunga analisi, né troppo specificamente in quanto può condurre ad un'insufficiente visione del problema;

#### **3.4.3.2 DEFINIZIONE DEL SISTEMA**

Un sistema è una struttura consistente in un numero di componenti di base interconnessi, funzionanti in modo tale che una missione assegnata possa compiersi, date le condizioni iniziali ed i vincoli imposti.

I sistemi possono essere rappresentati con diagrammi a blocchi. Un diagramma a blocchi consiste in un insieme di blocchi (rettangoli) che rappresentano i vari componenti del sistema considerato, collegati fra loro a seconda delle loro varie interconnessioni. A titolo esemplificativo, nella figura sottostante sono illustrati due diagrammi a blocchi (con rispettiva rappresentazione ad albero) relativi a due semplici sistemi formati ciascuno da due componenti operanti in serie ed in parallelo:

- il **gate OR** rappresenta una struttura in serie;
- il **gate AND** rappresenta una struttura parallela;



A ciascun blocco, rappresentativo di un dato componente, viene associato uno di due possibili stati (eventi): stato 0/UP (componente funzionante) oppure stato 1/DOWN (componente guasto). Quindi sono applicate le procedure dell'algebra booleana (chiamata propriamente anche algebra degli eventi).

Mediante queste procedure le operazioni di intersezione ( $\cap$ ) e di unione ( $\cup$ ) di due eventi associati a due blocchi corrispondono all'evento che risulta dalla loro connessione in serie (disgiunzione OR) o in parallelo (congiunzione AND), rispettivamente.

Basandoci su questa interpretazione, possiamo quindi dire che un sistema rappresentato da un diagramma a blocchi sarà funzionante (UP) se in tale schema esiste almeno un percorso continuo tra l'ingresso (input) e l'uscita (output).

Le intersezioni booleane che soddisfano questa condizione di continuità tra input ed output di un diagramma a blocchi sono chiamate "path sets". Quindi possiamo dire che se il Top Event coincide con il malfunzionamento di una parte dell'impianto, il *path set* è quello stato del sistema che porta al corretto funzionamento. Un *minimal path set* è un path set che non comprende alcun altro path set.

Va notato che la Fault Tree Analysis è un metodo deduttivo, il che comporta una grande abilità da parte dell'analista soprattutto in queste fasi di conoscenza e studio del sistema in esame (identificazione dei modi di guasto dei componenti/sottoinsiemi che sono inclusi nell'analisi; valutazione dell'applicabilità di errori umani; possesso di precise informazioni circa le funzioni svolte da ogni parte del dispositivo, le condizioni operative reali, il tipo di utilizzo; conoscenza esatta della manutenibilità del sistema...).

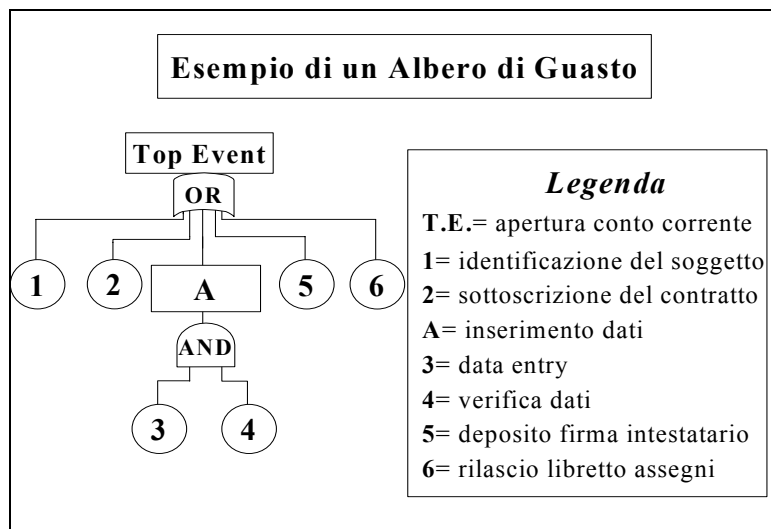
### **3.4.3.3 COSTRUZIONE DELL'ALBERO**

La costruzione dell'albero (solitamente manuale) implica la ricerca delle possibili cause che provocano il T.E. partendo dalle cause immediate fino ad arrivare alle cause elementari; vengono, cioè, esplicitate tutte le combinazioni di eventi di guasto ad un certo livello che sono necessarie a causare l'evento di guasto di livello superiore e ciò si ripete partendo dal T.E. di sistema fino a giungere al livello di dettaglio voluto.

Un operatore logico (AND, OR...) combina questi eventi in modo da individuare l'espressione booleana relativa al T.E., cosa che si riflette graficamente nell'uso di una struttura ad albero e di una simbologia appropriata.

Si riporta in seguito un semplice esempio di albero di guasto. In questo caso l'evento T.E. corrisponde all'apertura di un conto corrente.





Come si può vedere dalla figura:

- Il nodo A si guasta solo se contemporaneamente si guastano anche il Basic Event 3 e il Basic Event 4, cioè “l’inserimento dei dati su supporti informatici” presenta un errore sia perché c’è stato un errore nella fase di “data entry”, sia perché c’è stato un errore nella “verifica dei dati”.
- Il Top Event è nello stato di non funzionamento se uno solo dei suoi sottounodi (1, 2, A, 5, 6) presenta un errore, cioè “l’apertura di un conto corrente” presenta uno stato di guasto se si è commesso un errore “nell’identificazione del soggetto richiedente”, oppure durante “la sottoscrizione del contratto”, o al momento “dell’inserimento dei dati” oppure durante “la firma dell’intestatario” (per esempio gli intestatari sono più di uno ma nel contratto è presente un’unica firma) oppure non viene “rilasciato al cliente il libretto degli assegni”.

Una volta che l’albero di guasto è stato costruito, la probabilità del *Top Event* può essere determinata calcolando la probabilità degli eventi intermedi partendo dal basso dell’albero e procedendo verso l’alto.

Nella FTA si considerano solamente i sistemi binari, ossia quei dispositivi in cui ogni componente, ed in ultima analisi anche l’intero sistema stesso, può assumere unicamente le due condizioni “*funzionante – non funzionante*”; in

letteratura esistono anche delle trattazioni riguardanti casi in cui è necessario definire eventi multistato.

### 3.4.3.4 VALUTAZIONE/ANALISI DELL'ALBERO

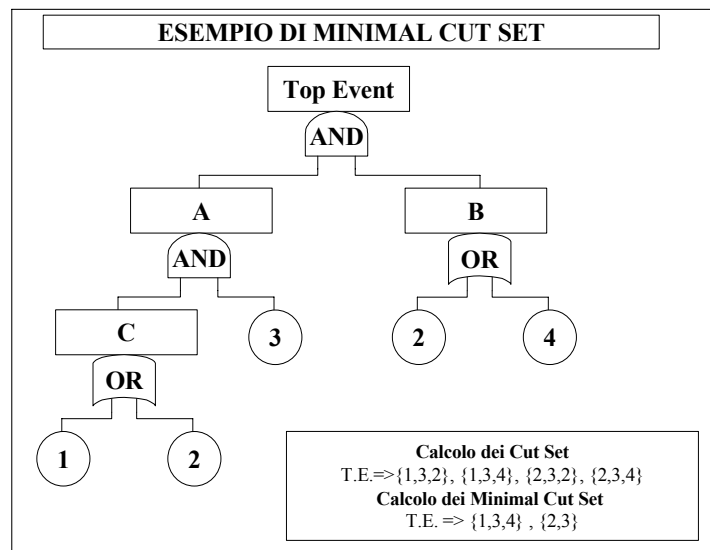
Questa fase ha lo scopo di determinare tutte le possibili cause del T.E., che possono essere singole o combinazioni di eventi primari.

Se non si ha informazione numerica circa la probabilità di accadimento dei singoli eventi si procede **all'analisi qualitativa (o logica)** degli alberi, che consiste principalmente nella rappresentazione analitica dell'albero dei guasti e nella determinazione/interpretazione dei Minimal Cut Sets. Se l'**analisi è quantitativa**, si ottiene anche una valutazione della probabilità di accadimento del T.E., dell'importanza dei componenti e dei M.C.S..

Le concatenazioni di eventi di base che causano un *Top Event* vengono indicate con il nome di *Cut-Set*.

Un Minimal Cut Set (M.C.S.) è definito come la più piccola combinazione di eventi base che, se accadono tutti, causano il Top Event.

Facciamo un esempio:



Dalla conoscenza dei M.C.S. per un albero di guasto, si ha la visione dei potenziali punti deboli del sistema, anche quando non è possibile calcolare la loro probabilità, infatti:

- Più sono i Minimal Cut Sets più sono i possibili insiemi di componenti/sottoinsiemi che possono provocare il T.E..
- L'ordine di un Minimal Cut Set rappresenta il numero di eventi base in esso contenuti. Più l'ordine del M.C.S è basso più il M.C.S. è importante, poiché sequenze corrispondenti ad un numero piccolo di guasti possono produrre il più grande contributo al guasto del sistema. Quindi i single failure (M.C.S. di ordine 1), se ci sono, sono i più pericolosi e vi è urgente richiesta di soluzione, di conseguenza i M.C.S. del secondo ordine sono più importanti di quelli del terzo ordine e così via. Le sequenze di ordine alto invece sono utili per cercare possibili cause comuni di eventi base incluse nello stesso cammino.

Quindi possiamo dire che l'importanza di un componente incrementa con il numero dei M.C.S. in cui esso appare e decrementa con il suo ordine. Inoltre il tipo di evento fornisce un'informazione utile per l'analisi di criticità. Ad esempio un evento relativo ad un errore umano può essere più frequente di uno relativo ad un guasto di un componente. Sulla base di tali criteri qualitativi è possibile elencare gli eventi in base alla loro importanza, al fine di individuare le zone più deboli del sistema (quelle dove sono localizzati i componenti più critici).

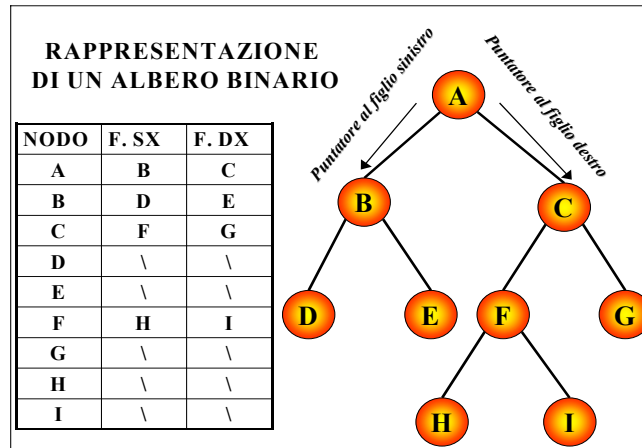
Per la determinazione dei M.C.S. sono stati sviluppati molteplici metodi: i più diffusi determinano i M.C.S. attraverso la sostituzione della definizione degli operatori a partire rispettivamente dal Top Event (analisi top-down) oppure dagli eventi primari (analisi bottom-up) e utilizzando le regole dell'algebra booleana.

## CAPITOLO 4

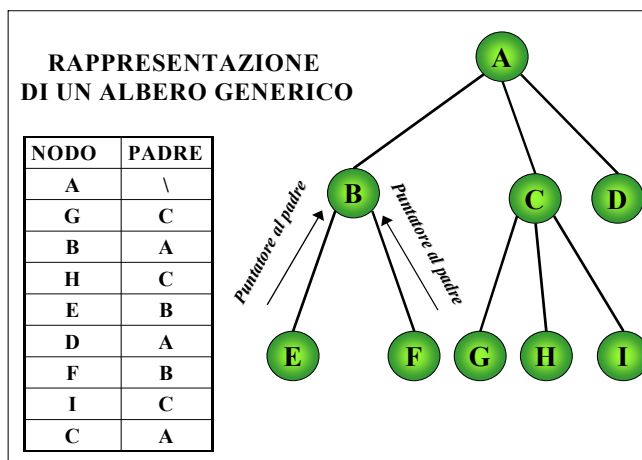
# APPLICAZIONE DELL'ORDINAMENTO SUGLI ALBERI DI GUASTO

In letteratura molto spesso si parla di visita di un albero o di ordinamento ma la maggior parte degli algoritmi presenti si riferiscono alle visite di alberi binari.

Gli alberi binari possono essere implementati mediante record di tre campi: R (la radice), S (puntatore al figlio sinistro), D (puntatore al figlio destro), come possiamo vedere dalla figura seguente:



Per quanto riguarda gli alberi generici invece, essi sono implementati mediante record a due campi: R (la radice) e P (puntatore al padre)



Gli algoritmi più semplici sono a volte, anche i meno efficienti; la loro virtù è la semplicità, tuttavia a volte occorre passare ad algoritmi più complessi per ottimizzare l'efficienza di un programma.

L'input che noi abbiamo a disposizione è un albero generico sottoforma matriciale che come unica informazione ci dà la relazione di padre-figlio per ogni nodo (come possiamo vedere dalla figura precedente), quindi non possiamo utilizzare gli algoritmi più semplici già presenti in letteratura ma dobbiamo creare dal principio, utilizzando il software Mathcad, un programma che calcoli il preordine della nostra matrice di dati, indipendentemente dal numero di nodi, dal numero di livelli e dal numero di grado di ogni nodo.

La proposta progettuale di Engineering consiste nella fornitura di soluzioni software riguardanti la consulenza e la gestione dei rischi operativi.

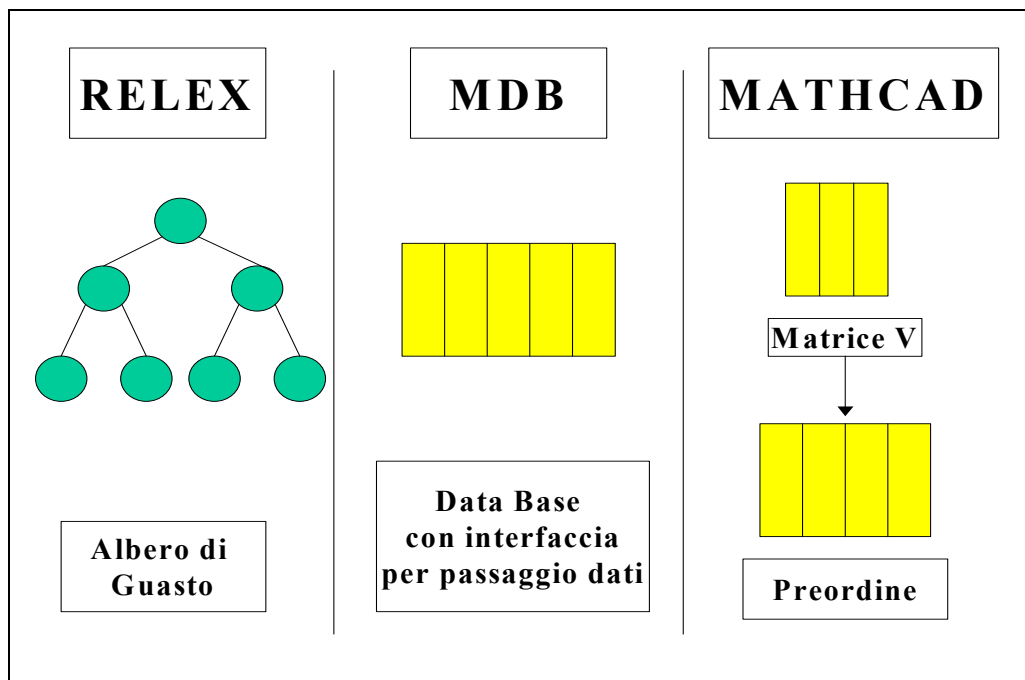
L'architettura globale del sistema è formata da:

- Un data base integrato processi/rischi/procedure operativo in ambiente ARIS; ARIS è un tool che permette, secondo diversi formalismi, di “mappare” i processi/attività di una banca. Tramite ARIS, leader in questo campo, è possibile rappresentare sequenze, volumi, eventi di rischio, controlli, responsabili, ecc... di tali procedure.
- Un data base contenente l'*Albero dei Guasti*, tramite l'utilizzo del software RELEX, tale software permette sia la creazione e gestione di Alberi di Guasto, sia la loro analisi (FTA). In ambiente RELEX gli alberi sono grafi dove chiunque, a prima vista, può individuarne la scala gerarchica, quello che vogliamo fare è trasformare questi grafi in tabelle senza perdere l'ordinamento dei nodi.
- Un data base che viene alimentato con dati provenienti da:
  - 1) Data base integrato *processi/rischi/procedure*;
  - 2) Dati e stime ottenuti attraverso il motore di calcolo;
  - 3) Data base dell'Albero di Guasto.

Il data base MDB viene alimentato da ARIS, RELEX, Mathcad, Access... e se analizziamo graficamente un Albero di Guasto possiamo vedere che le informazioni necessarie per rappresentarlo in modo univoco sono tre: il nome del nodo/informazione, il predecessore o padre del nodo e la tipologia del nodo. Per trasformare un Albero di Guasto da un grafico ad una tabella utilizziamo Access e in particolare Oprisk: inseriamo le tre informazioni che caratterizzano un albero in tre colonne e cerchiamo di trovare un modo per ordinare questi nodi in modo che la scala gerarchica si possa vedere anche da una singola tabella, questo per dare la possibilità a qualsiasi tipologia di cliente di distinguere velocemente e nel modo più semplice possibile la gerarchia di qualsiasi Albero di Guasto.

- Motore di calcolo Mathcad per la lavorazione sulle tabelle degli Alberi di Guasto e la generazione di report;

Lo schema logico che abbiamo seguito è quello riportato nella figura seguente:



## 4.1 PERCHE' E' IMPORTANTE L'ORDINAMENTO IN UN ALBERO DI GUASTO?

I rischi operativi sono rappresentati da alberi di guasto ed utilizzando Oprisk si possono distinguere subito le relazioni gerarchiche che intercorrono tra i nodi di un albero, perciò anche una persona esterna che non conosce la materia non avrà difficoltà a leggere questo tipo di dati.

Il problema sussiste quando una serie di informazioni, relazionate tra loro, vengono rappresentate in forma tabellare e quindi la “parentela” dei nodi/informazioni viene persa; per risolvere questo tipo di problema abbiamo iniziato a lavorare in Mathcad, allo scopo di creare un algoritmo che ordini la nostra tabella in modo che chiunque al primo sguardo possa distinguere il Top Event e le varie relazioni che legano i nodi intermedi e finali al nodo radice.

La tabella, con i dati di input, dalla quale partiremo sarà di questo tipo (ovviamente per comodità utilizzeremo una quantità minima di dati):

fname	ftparent	ftype
cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP		2
b13eb4b7-9596-47c2-9b8a-4b72fc77af4d - Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward) 10	cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP	2
1dc3be61-20a2-46ec-8f97-1e22dcdfffc3 - Invio conferme (derivati su cambi OTC)	cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP	2
01d374c2-246b-401a-bcc1-12898ae5497b - Contatto (derivati su cambi OTC)	cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP	2
1c33cd90-3e18-4bb8-bea4-44cb1b1e0405 - Esercizio opzioni e gestione scadenze futures (derivati su cambi OTC)	cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP	2
33ffa3e8-6fb2-4135-8a02-556ae3d6a24b - Input e Validazione (derivati su cambi OTC- Spot/Forward)	cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP	2
483c118c-1de2-405d-8cd9-393f0211199f - Ricezione flussi contabili (MXRATES) 15	cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP	2
e0186b91-8f72-4c8e-86e9-1d74886985e4 - Esecuzione (derivati su cambi OTC)	cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP	2
7f33486c-ac55-4697-ae28-dd2b0922419e - Spunta conferme e riconciliazione (derivati su cambi OTC)	cf7a3b33-8c41-11d7-b19e-0008c7ff3254 - DERIVATI SU CAMBI OTC CP	2
632680ef-e4bd-49b3-95a1-0dff0743a6a6 - VERIFICARE la conferma di MxG2000	7f33486c-ac55-4697-ae28-dd2b0922419e - Spunta conferme e riconciliazione (derivati su cambi OTC)	2

*Ftname*: corrisponde ad una stringa di caratteri e rappresenta il “nome” di ogni nodo, questo “nome” è formato da due parti: un codice e una seconda parte descrittiva.

*Ftparent*: corrisponde al nodo che precede l'*Ftname*, nel caso in cui l'*Ftname* corrisponda al Top Event il suo *Ftparent* sarà pari a zero in quanto il Top Event è la radice dell'albero.

*Fttype*: rappresenta la tipologia del nodo in prima colonna; viene assegnato il valore 2 al Top Event e a tutti i nodi interni (cioè i nodi che hanno almeno un successore) mentre si assegna il valore 7 ai Basic Event (le foglie dell'albero).

Da una tabella di questo tipo non sono molte le informazioni che si possono apprendere a prima vista, sappiamo qual è il rapporto padre-figlio ma non conosciamo l'intera scala gerarchica dell'albero di guasto.

Quello che si vuole fare con l'algoritmo che mostreremo tra poco è proprio questo: individuare subito la scala gerarchica di un albero di guasto indipendentemente la suo numero di nodi.

Il risultato finale, in riferimento alla tabella precedente, sarà questo:

	0	1	2	3	4
0	"-----"	CAMBI OTC CP"	0	2	0
1	"-----"	su cambi OTC)"	CAMBI OTC CP"	2	1
2	"-----"	e linee di credito"	su cambi OTC)"	2	2
3	"---"	nsimento ctp (a)"	e linee di credito"	7	3
4	"-----"	su cambi OTC)"	CAMBI OTC CP"	2	1
5	"-----"	zio su MxG2000"	su cambi OTC)"	2	2
6	"---"	ne in the money"	zio su MxG2000"	7	3
7	"-----"	su cambi OTC)"	CAMBI OTC CP"	2	1
8	"-----"	ma di MxG2000"	su cambi OTC)"	2	2
9	"---"	prez. conferma)"	ma di MxG2000"	7	3

Ora possiamo dire che anche da una tabella si può visualizzare l'ordinamento di ogni singolo nodo, sia guardando l'identazione della colonna zero, sia guardando i livelli della colonna quattro.



Per rendere ancora più comprensibile la gerarchia dell'albero possiamo togliere il codice dall'Ftname ed unire la descrizione con l'identatura ottenendo:

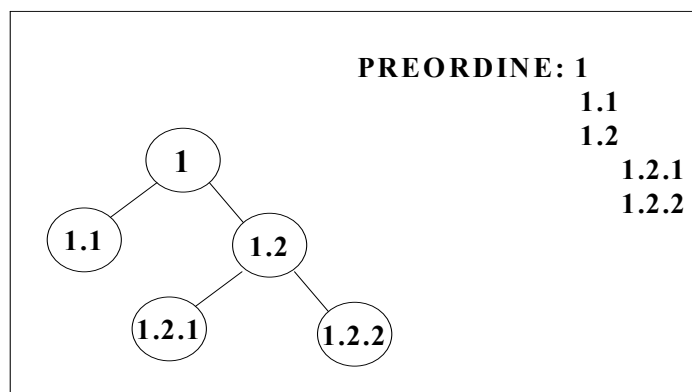
	0
0	" DERIVATI SU CAMBI OTC CP _____"
1	" Contatto (derivati su cambi OTC) _____"
2	" RICHIEDERE la verifica delle linee di credito _____"
3	" Mancato rispetto disposizioni interne: mancata verifica censimento ctp (a) _____"
4	" Esercizio opzioni e gestione scadenze futures (derivati su cambi OTC) _____"
5	" INSERIRE i dati relativi all esercizio su MxG2000 _____"
6	" Mancato esercizio di un opzione in the money _____"
7	" Invio conferme (derivati su cambi OTC) _____"
X2 = 8	" VERIFICARE la conferma di MxG2000 _____"
9	" Errato input manuale nei sistemi (correz. conferma) _____"

L'attraversamento o visita di un albero può essere di diversi tipi, per i nostri scopi useremo il *preordine* che corrisponde ad una visita in profondità dove ogni nodo è inserito prima dei suoi discendenti.

Questo tipo di ordinamento ci fa pensare all'indice di un libro dove:

- ai capitoli corrispondono i Top Event di ogni albero;
- ai sottocapitoli corrispondono i figli del Top Event;
- ai paragrafi corrispondono i nipoti del Top Event;
- e così via.

Quanto appena detto si può vedere anche da questo semplice esempio:



## 4.2 CREAZIONE DELL'ALGORITMO

Il primo passo è quello di rappresentare un albero di guasto in forma matriciale in quanto Mathcad lavora utilizzando le matrici.

Per non perdere informazioni da una forma ad albero ad una matrice dobbiamo seguire uno schema univoco, ogni albero in forma matriciale deve avere:

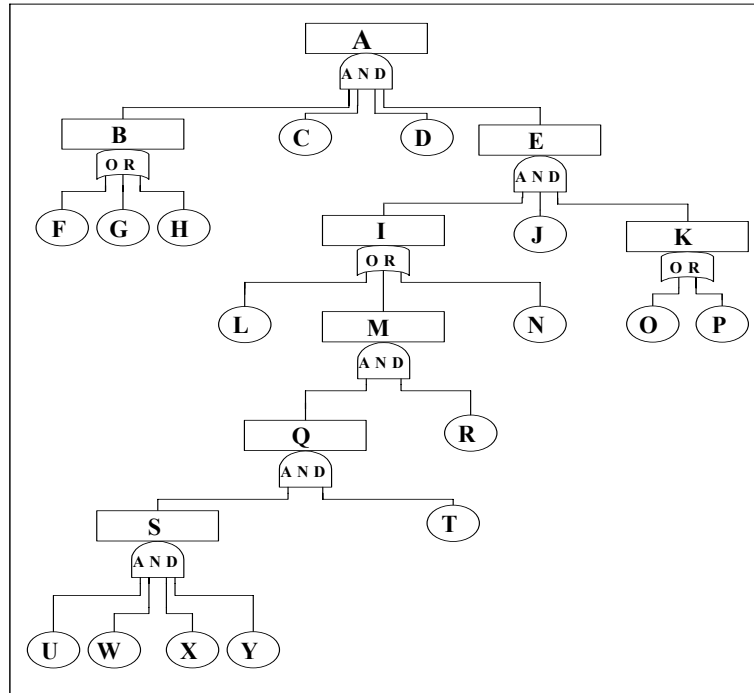
- In prima colonna il nome di ogni componente dell'albero che chiameremo FTNAME;
- In seconda colonna il predecessore di ogni componente che chiameremo FTPARENT;
- In terza colonna il tipo di gate di ogni componente (Basic Event, OR gate, AND gate...) che chiameremo FTTYPE;

Avremo quindi una matrice  $n \times 3$  dove  $n$  è il numero di componenti/nodi e per ognuno di essi mostra le tre informazioni presenti nelle 3 colonne.

Per distinguere il Top Event (la radice dell'albero) dai nodi interni e dalle foglie osserviamo l'indicazione del suo FTPARENT, questa riga risulterà vuota in quanto lo dice la definizione stessa di Top Event (la radice di un albero non ha predecessori).

### 4.2.1 ESEMPIO DI UN ALBERO DI GUASTO

La figura successiva mostra un generico albero di guasto, per semplicità sono state utilizzate le lettere dell'alfabeto per rappresentare sia Basic Event che nodi intermedi, nella realtà l'FTNAME è composto da stringhe di caratteri che identificano ogni nodo.



Come possiamo vedere questo algoritmo ha 24 componenti quindi la nostra matrice avrà 24 righe e come detto prima 3 colonne.

Indicheremo con:

- $i$  la  $i$ -esima riga dove  $0 \leq i \leq n-1$ ;
- $j$  la  $j$ -esima colonna dove  $0 \leq j \leq 2$ ;
  - con  $j=0$  indichiamo FTNAME;
  - con  $j=1$  indichiamo FTPARENT;
  - con  $j=2$  indichiamo FTTYPE;

quindi l'elemento  $V_{i,j}$  sarà relativo alla riga  $i$ -esima e alla colonna  $j$ -esima.

L'inserimento dei nodi nella matrice sarà casuale e per il nostro scopo non è fondamentale sapere se il gate è un OR oppure un AND, ciò che ci interessa è se il nodo è una foglia (Basic Event) oppure un nodo interno; quindi useremo due valori che facciano questa distinzione:

- 7 per i nodi foglia;
- 2 per i nodi interni;

$$V := \begin{pmatrix} "G" & "B" & 7 \\ "T" & "Q" & 7 \\ "X" & "S" & 7 \\ "O" & "K" & 7 \\ "P" & "K" & 7 \\ "J" & "E" & 7 \\ "I" & "E" & 2 \\ "L" & "T" & 7 \\ "F" & "B" & 7 \\ "Y" & "S" & 7 \\ "R" & "M" & 7 \\ "H" & "B" & 7 \\ "D" & "A" & 7 \\ "W" & "S" & 7 \\ "Q" & "M" & 2 \\ "E" & "A" & 2 \\ "N" & "T" & 7 \\ "M" & "T" & 2 \\ "U" & "S" & 7 \\ "S" & "Q" & 2 \\ "A" & 0 & 2 \\ "B" & "A" & 2 \\ "K" & "E" & 2 \\ "C" & "A" & 7 \end{pmatrix}$$

Otterremo così la matrice  $V$ :

La matrice ottenuta sarà di questo tipo:

nella riga zero c'è il nodo con FTNAME "G" che è collegato al nodo "B" ed è un nodo foglia in quanto in colonna due c'è il numero 7;

nella riga uno troviamo il nodo con FTNAME "T" che è collegato al nodo "Q" ed è anch'esso un nodo foglia in quanto in colonna due vi è il numero 7; e così via.

Il nodo padre è nella 20esima riga e questo possiamo dirlo in quanto nella posizione  $V_{20,1}$  c'è lo zero.

Ora che abbiamo l'albero scritto in forma matriciale, è possibile applicare gli algoritmi per il calcolo dell'ordinamento:

- Calcolo del livello di ogni nodo:
  - Algoritmo Z per il calcolo del Top Event;
  - Algoritmo Z1 per il calcolo dei livelli di ogni nodo;
- Calcolo dell'ordinamento dell'Albero di Guasto:
  - Algoritmo P per il calcolo del cammino di ogni nodo (da qualsiasi nodo dell'albero al Top Event);
  - Algoritmo M di semplificazione;
  - Algoritmo S per la trasposizione della matrice M, il capovolgimento delle righe e successiva trasposizione;

- Algoritmo Q per lo spostamento delle celle vuote da sinistra a destra;
- Algoritmo L1 per la trasformazione di tutti i valori della matrice in stringhe;
- Algoritmo H2 per la concatenazione degli elementi di ogni riga;
- Algoritmo Z per unire alla matrice L1 la colonna H2;
- Algoritmo Z2 per ordinare la matrice Z in base alla colonna H2;
- Algoritmo Z3 per la trasposizione della matrice M, il capovolgimento delle righe e successiva trasposizione;
- Algoritmo Q per lo spostamento delle celle vuote da sinistra a destra;
- Algoritmo Q6 di semplificazione;
- Algoritmo W6 per l'indentatura;
- Algoritmo X per calcolare la differenza tra il massimo degli elementi che formano le descrizioni di ogni nodo e gli elementi di ogni descrizione;
- Algoritmo X1 per rendere le descrizioni della stessa lunghezza;
- Algoritmo X2 per concatenare X1 con l'indentatura di W6;

#### 4.2.1.1 CALCOLO DEI LIVELLI DI UN ALBERO/MATRICE

Il primo passo è quello di trovare il nodo padre ed estrarlo dalla matrice  $V$ , a questa riga aggiungiamo in ultima colonna il livello del nodo e già sappiamo che il livello del Top Event sarà zero.

$$Z = ("A" 0 2 0)$$

Il passo successivo è di costruire un algoritmo che metta nella prima riga della nuova matrice la riga del Top Event, poi il programma analizza tutte le righe della matrice  $V$  e quando trova che l'FTPARENT del nodo esaminato è lo stesso dell'FTNAME del nodo padre allora la riga viene inserita e in quarta colonna il

livello aumenta di 1. Quando sono stati inseriti tutti i nodi con livello uno si riesaminano di nuovo le righe della matrice  $V$  ma questa volta gli FTPARENT non vengono confrontati con l'FTNAME del Top Event ma con l'FTNAME dei nodi interni (quindi in terza colonna non devono avere 7) già inseriti in matrice che hanno livello 1; a questi nodi in quarta colonna viene inserito il livello aumentato di uno. Poi le righe della matrice  $V$  vengono riesaminate e si confrontano con i nodi di livello 2 e così via. L'algoritmo si ferma quando ha analizzato tutti i nodi della matrice  $V$  e li ha riportati in una nuova matrice ordinandoli per livello. Il risultato è la matrice seguente:

	0	1	2	3
0	"A"	0	2	0
1	"D"	"A"	7	1
2	"E"	"A"	2	1
3	"B"	"A"	2	1
4	"C"	"A"	7	1
5	"J"	"E"	7	2
6	"I"	"E"	2	2
7	"K"	"E"	2	2
8	"G"	"B"	7	2
9	"F"	"B"	7	2
10	"H"	"B"	7	2
11	"L"	"I"	7	3
12	"N"	"I"	7	3
13	"M"	"I"	2	3
14	"O"	"K"	7	3
15	"P"	"K"	7	3
16	"R"	"M"	7	4
17	"Q"	"M"	2	4
18	"T"	"Q"	7	5
19	"S"	"Q"	2	5
20	"X"	"S"	7	6
21	"Y"	"S"	7	6
22	"W"	"S"	7	6
23	"U"	"S"	7	6

#### 4. 2.1.2 ORDINAMENTO DELL'ALBERO DI GUASTO

Il primo algoritmo relativo a questa parte calcola una matrice dove si possono vedere i cammini di ogni nodo, nelle prime due colonne vengono riportate le prime due colonne della matrice  $ZI$  e a mano a mano che si prosegue verso destra

si vedranno tutti gli antenati del nodo di sinistra e, nelle ultime due colonne vengono riportati gli FTTYPE e i LIVELLI di ogni nodo in prima colonna.

Analizziamo ora l'algoritmo, si parte dalla matrice  $ZI$  e si studiano i livelli di ogni nodo:

- se il livello è  $\leq 1$  allora nella colonna 2 viene riportato zero;
- se il livello è  $= 2$  allora nella colonna 2 viene riportato il nodo padre;
- se il livello è  $\geq 3$  allora si confronta per ogni  $i \in 0 \dots n-1$  e per ogni  $s \in 0 \dots n-1$  l'elemento in posizione  $ZI_{i,j+1}$  con l'elemento  $ZI_{s,j}$  e se sono uguali inserisco in  $P_{i,j+2}$  l'elemento  $ZI_{s,j+1}$ , il programma si ferma quando tutti i nodi col più alto livello sono stati analizzati e inseriti nella nuova matrice.

La matrice che risulta è la seguente:

	0	1	2	3	4	5	6	7	8
0	"A"	0	0	0	0	0	0	2	0
1	"D"	"A"	0	0	0	0	0	7	1
2	"E"	"A"	0	0	0	0	0	2	1
3	"B"	"A"	0	0	0	0	0	2	1
4	"C"	"A"	0	0	0	0	0	7	1
5	"J"	"E"	"A"	0	0	0	0	7	2
6	"I"	"E"	"A"	0	0	0	0	2	2
7	"K"	"E"	"A"	0	0	0	0	2	2
8	"G"	"B"	"A"	0	0	0	0	7	2
9	"F"	"B"	"A"	0	0	0	0	7	2
10	"H"	"B"	"A"	0	0	0	0	7	2
P = 11	"L"	"I"	"E"	"A"	0	0	0	7	3
12	"N"	"I"	"E"	"A"	0	0	0	7	3
13	"M"	"I"	"E"	"A"	0	0	0	2	3
14	"O"	"K"	"E"	"A"	0	0	0	7	3
15	"P"	"K"	"E"	"A"	0	0	0	7	3
16	"R"	"M"	"I"	"E"	"A"	0	0	7	4
17	"Q"	"M"	"I"	"E"	"A"	0	0	2	4
18	"T"	"Q"	"M"	"I"	"E"	"A"	0	7	5
19	"S"	"Q"	"M"	"I"	"E"	"A"	0	2	5
20	"X"	"S"	"Q"	"M"	"I"	"E"	"A"	7	6
21	"Y"	"S"	"Q"	"M"	"I"	"E"	"A"	7	6
22	"W"	"S"	"Q"	"M"	"I"	"E"	"A"	7	6
23	"U"	"S"	"Q"	"M"	"I"	"E"	"A"	7	6

Al passo successivo vogliamo creare una nuova colonna alla matrice  $P$  che contenga una stringa unica dove si possa vedere il cammino di ogni nodo, dal

nodo padre al nodo foglia, una volta fatto questo, il problema dell'ordinamento (preordine) è risolto in quanto in Mathcad c'è un comando che ordina una matrice in base ad una colonna.

Prima di tutto lavoriamo con una matrice formata solo da FTNAME e FTPARENT esclusivamente per comodità, quindi togliamo le ultime due colonne da  $P$ .

La nuova matrice  $M$  ha in prima colonna i nodi dell'albero e nelle tuple successive i relativi antenati, la nostra idea è quella di pensare alla matrice come se fosse un libro avente un unico capitolo (Top Event) e molti sottocapitoli (nodi intermedi e foglie) che poi ordineremo, quindi dobbiamo stravolgere la matrice  $M$  in modo da avere a sinistra il nodo padre e mano a mano che ci si sposta verso destra i relativi discendenti fino ad arrivare alle foglie. Questo si può fare utilizzando il seguente comando:

$S := reverse(M^T)^T$  dove  $M$  è la matrice  $P$  senza le ultime due colonne;

il comando "reverse" rovescia letteralmente una matrice, cioè l'ultima riga diventa la prima, la penultima diventa la seconda e così via.

La matrice ottenuta è la seguente:

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	"A"
1	0	0	0	0	0	"A"	"D"
2	0	0	0	0	0	"A"	"E"
3	0	0	0	0	0	"A"	"B"
4	0	0	0	0	0	"A"	"C"
5	0	0	0	0	"A"	"E"	"J"
6	0	0	0	0	"A"	"E"	"I"
7	0	0	0	0	"A"	"E"	"K"
8	0	0	0	0	"A"	"B"	"G"
9	0	0	0	0	"A"	"B"	"F"
10	0	0	0	0	"A"	"B"	"H"
11	0	0	0	"A"	"E"	"I"	"L"
12	0	0	0	"A"	"E"	"I"	"N"
13	0	0	0	"A"	"E"	"I"	"M"
14	0	0	0	"A"	"E"	"K"	"O"
15	0	0	0	"A"	"E"	"K"	"P"
16	0	0	"A"	"E"	"I"	"M"	"R"
17	0	0	"A"	"E"	"I"	"M"	"Q"
18	0	"A"	"E"	"I"	"M"	"Q"	"T"
19	0	"A"	"E"	"I"	"M"	"Q"	"S"
20	"A"	"E"	"I"	"M"	"Q"	"S"	"X"
21	"A"	"E"	"I"	"M"	"Q"	"S"	"Y"
22	"A"	"E"	"I"	"M"	"Q"	"S"	"W"
23	"A"	"E"	"I"	"M"	"Q"	"S"	"U"



Dobbiamo lavorare ancora un po' sulla matrice  $S$ , tutti gli zeri si devono spostare a destra e questo viene fatto con l'algoritmo  $Q$ ; questo algoritmo per ogni  $i \in 0 \dots \text{rows}(S)-1$  e per ogni  $j \in 0 \dots \text{cols}(S)-1$  analizza la riga  $i$  della matrice  $S$  e inserisce il contenuto delle tuple, relative ad essa, diverse da zero, nella colonna  $j$  della nuova matrice, il programma termina quando vengono analizzate tutte le righe di  $S$ . Il risultato è la matrice seguente:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"	"A"
1	0	"D"	"E"	"B"	"C"	"E"	"E"	"E"	"B"	"B"	"B"	"E"	"E"	"E"	"E"	"E"	"E"	"E"	"E"	"E"	"E"	"E"	"E"	"E"
2	0	0	0	0	0	"J"	"I"	"K"	"G"	"F"	"H"	"I"	"I"	"I"	"K"	"K"	"I"	"I"	"I"	"I"	"I"	"I"	"I"	"I"
3	0	0	0	0	0	0	0	0	0	0	0	"L"	"N"	"M"	"O"	"P"	"M"	"M"	"M"	"M"	"M"	"M"	"M"	"M"
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	"R"	"Q"	"Q"	"Q"	"Q"	"Q"	"Q"	"Q"
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	"T"	"S"	"S"	"S"	"S"	"S"
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	"X"	"Y"	"W"	"U"

facendo la trasposta di  $Q$  ottengo:  $Q := Q^T$

	0	1	2	3	4	5	6
0	"A"	0	0	0	0	0	0
1	"A"	"D"	0	0	0	0	0
2	"A"	"E"	0	0	0	0	0
3	"A"	"B"	0	0	0	0	0
4	"A"	"C"	0	0	0	0	0
5	"A"	"E"	"J"	0	0	0	0
6	"A"	"E"	"I"	0	0	0	0
7	"A"	"E"	"K"	0	0	0	0
8	"A"	"B"	"G"	0	0	0	0
9	"A"	"B"	"F"	0	0	0	0
10	"A"	"B"	"H"	0	0	0	0
11	"A"	"E"	"I"	"L"	0	0	0
12	"A"	"E"	"I"	"N"	0	0	0
13	"A"	"E"	"I"	"M"	0	0	0
14	"A"	"E"	"K"	"O"	0	0	0
15	"A"	"E"	"K"	"P"	0	0	0
16	"A"	"E"	"I"	"M"	"R"	0	0
17	"A"	"E"	"I"	"M"	"Q"	0	0
18	"A"	"E"	"I"	"M"	"Q"	"T"	0
19	"A"	"E"	"I"	"M"	"Q"	"S"	0
20	"A"	"E"	"I"	"M"	"Q"	"S"	"X"
21	"A"	"E"	"I"	"M"	"Q"	"S"	"Y"
22	"A"	"E"	"I"	"M"	"Q"	"S"	"W"
23	"A"	"E"	"I"	"M"	"Q"	"S"	"U"

Per effettuare l'ordinamento in Mathcad ogni valore deve essere uno scalare o una stringa allora inseriamo gli apici nelle celle vuote, inoltre possiamo ora aggiungere le due colonne che prima avevamo tolto, FTTYPE e LIVELLO. A questo punto possiamo creare una nuova colonna che contenga l'intero cammino di ogni nodo.

In primo luogo concateniamo le ultime due colonne e successivamente concateniamo ad esse tutte le colonne della matrice  $Q$ , per esempio l'ultimo elemento della nuova colonna sarà composto così:

("SUAEIMQSU")

Dato che la concatenazione delle prime due colonne non ci interessa la togliamo usando il comando per le stringhe "*substr(a,b,c)*" dove  $a$  è la matrice alla quale dobbiamo togliere  $b$ , e  $b$  è il numero di caratteri che formano la stringa in posizione  $Q_{i,5}$  + il numero di caratteri che formano la stringa in posizione  $Q_{i,6}$  e  $c$  è la lunghezza massima che potrà avere la nuova stringa così composta.

La matrice ottenuta è:

	0	1	2	3	4	5	6	7	8	9
0	"A"	"0"	"0"	"0"	"0"	"0"	"0"	2	0	"A00000"
1	"A"	"D"	"0"	"0"	"0"	"0"	"0"	7	1	"AD00000"
2	"A"	"E"	"0"	"0"	"0"	"0"	"0"	2	1	"AE00000"
3	"A"	"B"	"0"	"0"	"0"	"0"	"0"	2	1	"AB00000"
4	"A"	"C"	"0"	"0"	"0"	"0"	"0"	7	1	"AC00000"
5	"A"	"E"	"J"	"0"	"0"	"0"	"0"	7	2	"AEJ0000"
6	"A"	"E"	"I"	"0"	"0"	"0"	"0"	2	2	"AEI0000"
7	"A"	"E"	"K"	"0"	"0"	"0"	"0"	2	2	"AEK0000"
8	"A"	"B"	"G"	"0"	"0"	"0"	"0"	7	2	"ABG0000"
9	"A"	"B"	"F"	"0"	"0"	"0"	"0"	7	2	"ABF0000"
10	"A"	"B"	"H"	"0"	"0"	"0"	"0"	7	2	"ABH0000"
11	"A"	"E"	"I"	"L"	"0"	"0"	"0"	7	3	"AEIL000"
12	"A"	"E"	"I"	"N"	"0"	"0"	"0"	7	3	"AEIN000"
13	"A"	"E"	"I"	"M"	"0"	"0"	"0"	2	3	"AEIM000"
14	"A"	"E"	"K"	"O"	"0"	"0"	"0"	7	3	"AEKO000"
15	"A"	"E"	"K"	"P"	"0"	"0"	"0"	7	3	"AEKP000"
16	"A"	"E"	"I"	"M"	"R"	"0"	"0"	7	4	"AEIMR00"
17	"A"	"E"	"I"	"M"	"Q"	"0"	"0"	2	4	"AEIMQ00"
18	"A"	"E"	"I"	"M"	"Q"	"T"	"0"	7	5	"AEIMQT0"
19	"A"	"E"	"I"	"M"	"Q"	"S"	"0"	2	5	"AEIMQS0"
20	"A"	"E"	"I"	"M"	"Q"	"S"	"X"	7	6	"AEIMQSX"
21	"A"	"E"	"I"	"M"	"Q"	"S"	"Y"	7	6	"AEIMQSY"
22	"A"	"E"	"I"	"M"	"Q"	"S"	"W"	7	6	"AEIMQSW"
23	"A"	"E"	"I"	"M"	"Q"	"S"	"U"	7	6	"AEIMQSU"

Ora possiamo riordinare la matrice  $Z$  in base all'ultima colonna, utilizzando il comando " $csort(a,b)$ " dove  $a$  è la matrice da ordinare (nel nostro caso  $Z$ ) e  $b$  è la colonna che ordina tale matrice (nel nostro caso la numero 9); otteniamo così un preordine. Per rendere a prima vista la matrice meno complessa cerchiamo di semplificarla, ricollochiamo a sinistra tutti i nodi dell'albero e i rispettivi antenati a destra utilizzando gli algoritmi visti in precedenza, poi eliminiamo le colonne centrali in modo da avere:

- in prima colonna gli FTNAME dei nodi nel nuovo ordine;
- in seconda colonna gli FTPARENT dei nodi nel nuovo ordine;
- in terza colonna gli FTTYPE dei nodi nel nuovo ordine;
- in quarta colonna i LIVELLI dei nodi nel nuovo ordine.

La matrice finale che mostra la matrice  $V$  in preordine è la seguente:

	0	1	2	3
0	"A"	0	2	0
1	"B"	"A"	2	1
2	"F"	"B"	7	2
3	"G"	"B"	7	2
4	"H"	"B"	7	2
5	"C"	"A"	7	1
6	"D"	"A"	7	1
7	"E"	"A"	2	1
8	"I"	"E"	2	2
9	"L"	"I"	7	3
10	"M"	"I"	2	3
11	"Q"	"M"	2	4
12	"S"	"Q"	2	5
13	"U"	"S"	7	6
14	"V"	"S"	7	6
15	"X"	"S"	7	6
16	"Y"	"S"	7	6
17	"T"	"Q"	7	5
18	"R"	"M"	7	4
19	"N"	"I"	7	3
20	"J"	"E"	7	2
21	"K"	"E"	2	2
22	"O"	"K"	7	3
23	"P"	"K"	7	3

L'ordine in profondità che stavamo cercando è dato dalla prima colonna perciò, facendo un esempio, la nostra visita inizierà dal nodo padre "A" per poi passare a un suo figlio "B" e a un figlio di "B" se esiste, nel nostro caso "F", se "F"

avesse avuto dei figli, a questo punto la matrice  $Q_6$  ne avrebbe segnalato la presenza ma dato che nella nostra matrice  $V$  il nodo "F" è una foglia allora mettiamo i suoi fratelli, i fratelli di "F" non hanno figli e quindi passiamo ai fratelli di "B" e così via.

Dalla matrice  $Q_6$  è un po' complicato vedere d'impatto l'ordinamento del nostro albero iniziale, allora cerchiamo di aggiungere una colonna che renda più veloce la comprensione dei livelli dell'albero, anche per chi non conosce approfonditamente la materia.

"Repeat(p,n)" è un piccolo programma che in Mathcad ripete stringhe di caratteri; quindi se il valore che voglio copiare in relazione ai livelli è  $p="---"$ , in prima colonna copio  $p$ :

- 1 volta se il livello del nodo è quello massimo;
- un numero di volte pari al (massimo dei livelli+1) se il livello del nodo è zero;
- $c+1$  volte per  $c \in 1 \dots (\text{massimo dei livelli}+1)$  se il livello del nodo è (massimo dei livelli- $c$ )

La matrice che otteniamo è la seguente:

	0	1	2	3	4
0	"-----"	"A"	0	2	0
1	"-----"	"B"	"A"	2	1
2	"-----"	"F"	"B"	7	2
3	"-----"	"G"	"B"	7	2
4	"-----"	"H"	"B"	7	2
5	"-----"	"C"	"A"	7	1
6	"-----"	"D"	"A"	7	1
7	"-----"	"E"	"A"	2	1
8	"-----"	"I"	"E"	2	2
9	"-----"	"L"	"I"	7	3
10	"-----"	"M"	"I"	2	3
11	"-----"	"Q"	"M"	2	4
12	"-----"	"S"	"Q"	2	5
13	"-----"	"U"	"S"	7	6
14	"-----"	"W"	"S"	7	6
15	"-----"	"X"	"S"	7	6
16	"-----"	"Y"	"S"	7	6
17	"-----"	"T"	"Q"	7	5
18	"-----"	"R"	"M"	7	4
19	"-----"	"N"	"I"	7	3
20	"-----"	"J"	"E"	7	2
21	"-----"	"K"	"E"	2	2
22	"-----"	"O"	"K"	7	3
23	"-----"	"P"	"K"	7	3

Come detto sopra la prima colonna della matrice  $W6$  aiuta la visione dell'ordinamento dell'albero.

A questo punto possiamo unire la colonna degli Ftname con la colonna dell'identatura ed invece di usare dati fittizi usiamo una matrice di dati reali.

Per semplicità concateniamo solo la descrizione dell'Ftname all'identatura ma prima di fare questo le descrizioni dovranno essere composte dalla stessa quantità di caratteri altrimenti l'identatura non avrà alcun senso.

Con la funzione " $strlen(substr(W6_{i,1},38,2000000))$ " viene calcolato il numero di caratteri di ogni descrizione:

- $strlen$  conta il numero di caratteri di una stringa
- $substr(a,b,c)$  toglie da  $a$  il numero  $b$  di caratteri e la nuova stringa sarà formata al massimo da  $c$  caratteri ( $b$  rappresenta il codice dell'Ftname).

Ottenuto il vettore contenente il numero di caratteri di ogni stringa calcoliamo la differenza tra il maggiore di questi numeri ed ogni dimensione di descrizione, quindi moltiplicando ognuno di questi numeri per un determinato simbolo, nel nostro caso  $p = \_$  e concatenando i risultati ad ogni descrizione relativa otterremo descrizioni della stessa lunghezza.

Ora si può aggiungere l'identatura ma c'è ancora un piccolo problema da risolvere: ossia ogni descrizione è formata da elementi che possono essere lettere maiuscole, lettere minuscole, simboli e numeri, e le dimensioni di ognuno di essi sono diverse, quindi dobbiamo trovare un tipo di carattere per cui ogni elemento ha la stessa dimensione, questo problema è risolto da "Andale Mono" come possiamo vedere dal risultato finale.

	0
0	" DERIVATI SU CAMBI OTC CP _____"
1	" Contatto (derivati su cambi OTC) _____"
2	" RICHIEDERE la verifica delle linee di credito _____"
3	" Mancato rispetto disposizioni interne: mancata verifica censimento ctp (a) ___--"
4	" Esercizio opzioni e gestione scadenze futures (derivati su cambi OTC) _____"
5	" INSERIRE i dati relativi all esercizio su MxG2000 _____"
6	" Mancato esercizio di un opzione in the money _____"
7	" Invio conferme (derivati su cambi OTC) _____"
X2 = 8	" VERIFICARE la conferma di MxG2000 _____"
9	" Errato input manuale nei sistemi (correz. conferma) _____"
10	" Input e Validazione (derivati su cambi OTC- Spot/Forward) _____"
11	" VALIDARE l operazione in MxG2000 _____"
12	" Mancata validazione dei deals nei sistemi di FO _____"
13	" INSERIRE i dati relativi all operazione nel foglio Excel _____"
14	" Errato input manuale (parametri operazione) _____"
15	" Ricezione flussi contabili (MXRATES) 15 _____"
16	" ATTIVARE la query di investigation in PERRY MASON 15 _____"
17	" Inefficienza procedurale: mancata identificazione resp. risoluzione errore 15---"
18	" ATTIVARE le procedure informatiche per l acquisizione dei dati in RECON* 15_-----"

### 4.3 COMPLESSITA' COMPUTAZIONALE

Per complessità si intende una funzione che associa il numero di dati da trattare al numero di operazioni da eseguire per trattare i suddetti dati.

Quello che vogliamo vedere con quest'analisi è come aumenta il tempo di calcolo dell'algoritmo all'aumentare dei dati del tracciato.

In questo paragrafo tratteremo l'analisi Empirica sulla complessità cioè l'analisi che si basa sul tempo di calcolo dell'algoritmo e per questa analisi abbiamo bisogno di un numero di campioni. Nel nostro caso abbiamo a disposizione 8 alberi di guasto in forma tabellare il cui numero di nodi/informazioni spazia da 54 a 854. Calcoliamo per ogni albero il tempo di calcolo dell'algoritmo d'ordinamento e analizziamo i risultati ottenuti.

### 4.3.1 SIMBOLI DI LANDAU

I simboli “O;  $\asymp$ ;  $\sim$ ; o” sono detti simboli di Landau. Essi rappresentano una notazione “stenografica”, molto utile nello studio del comportamento locale delle funzioni.

Siano dunque  $f$  e  $g$  due funzioni definite nell'intorno di  $c$ , tranne eventualmente nel punto  $c$ ; inoltre, sia  $g(x) \neq 0$  per  $x \neq c$ . Supponiamo che esista, finito o infinito,

$$\lim_{x \rightarrow c} \frac{f(x)}{g(x)} = l$$

Definizione:

- a) Se  $l$  è finito, diciamo che  $f$  è controllata da  $g$  per  $x$  tendente a  $c$ ; in tal caso, usiamo il simbolo

$$f = O(g) \quad x \rightarrow c$$

che leggiamo “ $f$  è O grande di  $g$  per  $x$  tendente a  $c$ ”. Tale proprietà può essere precisata meglio, distinguendo i seguenti casi:

- b) Se  $l$  è finito e  $\neq 0$ , diciamo che  $f$  è dello stesso ordine di grandezza di  $g$  per  $x$  tendente a  $c$ ; in tal caso, usiamo il simbolo

$$f \asymp g \quad x \rightarrow c$$

che leggiamo “ $f$  è equigrande con  $g$  per  $x$  tendente a  $c$ ”. Come caso particolare abbiamo:

- c) Se  $l=1$ , diciamo che  $f$  è equivalente a  $g$  per  $x$  tendente a  $c$ ; in tal caso, usiamo il simbolo

$$f \sim g \quad x \rightarrow c$$

- d) Infine, se  $l=0$ , diciamo che  $f$  è trascurabile rispetto a  $g$  per  $x$  tendente a  $c$ ; in tal caso, usiamo il simbolo

$$f = o(g) \quad x \rightarrow c$$

che leggiamo “ $f$  è  $O$  piccolo di  $g$  per  $x$  tendente a  $c$ ”.

Dalle definizioni precedenti resta escluso il caso in cui  $l$  sia infinito. Ma, se ciò accade, allora

$$\lim_{x \rightarrow c} \frac{g(x)}{f(x)} = \frac{1}{l} = 0$$

e dunque possiamo dire che  $g = o(f)$  per  $x \rightarrow c$

Alcune funzioni appaiono spesso nell'analisi di algoritmi ed in quest'analisi vengono usati specifici termini che mostreremo nella tabella seguente:

logaritmica	lineare	quadratica	polinomiale	esponenziale
$O(\log n)$	$O(n)$	$O(n^2)$	$O(n^k)$ ( $k \geq 1$ )	$O(a^n)$ ( $a > 1$ )

Facciamo qualche esempio:

Algoritmo	Tempo di calcolo	Ordine del tempo di calcolo
Algoritmo 1	$10n + 25n^2$	$O(n^2)$
Algoritmo 2	$20n \log n + 5n$	$O(n \log n)$
Algoritmo 3	$12 \log n + 0.05n$	$O(n)$
Algoritmo 4	$5n^2 \log n + 2n^3$	$O(n^3)$

Ora passiamo allo studio del nostro algoritmo per vedere a quale ordine del tempo di calcolo appartiene.



### 4.3.2 ANALISI EMPIRICA

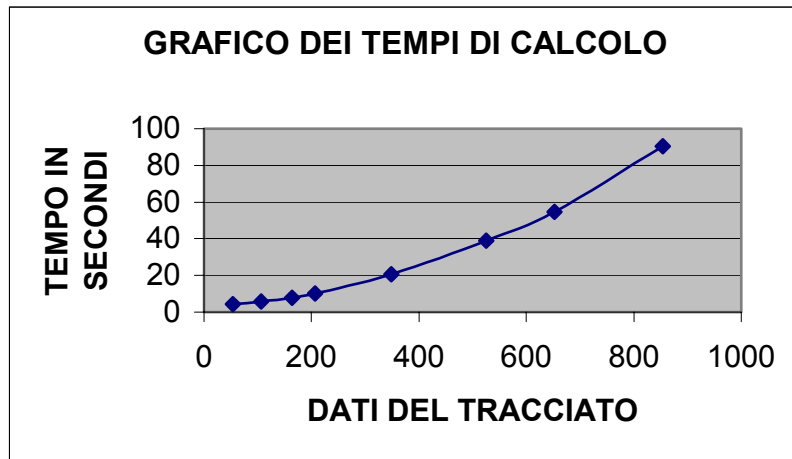
Studiamo il tempo di calcolo totale del nostro algoritmo, abbiamo a disposizione 8 tracciati (matrici) con una numerosità di nodi che va da 54 a 854.

La tabella che risulta dalle prove fatte è la seguente:

DATI	TEMPO IN SEC.
54	4.32
106	5.82
164	7.97
207	10.19
349	20.51
525	38.88
652	54.66
854	90.28

Osservando i risultati notiamo che se i dati aumentano anche il tempo aumenta, ora ci resta da vedere in che modo.

Rappresentiamo il grafico dei tempi di calcolo.



Dal grafico si può vedere che la complessità computazionale dell'algoritmo analizzato è polinomiale ma per esserne certi analizziamo la regressione.

**REGRESSIONE**

Supponiamo che esista una relazione di tipo quadratico fra la numerosità e il tempo di calcolo. Immaginiamo un modello di questo tipo:

$$T = C(1) \times n + C(2) \times n^2$$

Facciamo quindi una regressione tra i tempi di calcolo trovati con l'algoritmo e questo modello. I dati sono riportati nella seguente tabella dove in prima colonna figurano i tempi di calcolo, in seconda colonna la numerosità dei tracciati e in terza colonna il loro quadrato.

T	n	n <sup>2</sup>
4.32	54	2916
5.82	106	11236
7.97	164	26896
10.19	207	42849
20.51	349	121801
38.88	525	275625
54.66	652	425104
90.28	854	729316

Il risultato della regressione è riportato nella seguente tabella:

OUTPUT RIEPILOGO

<i>Statistica della regressione</i>	
R multiplo	0.99883554
R al quadrato	0.99767243
R al quadrato corretto	0.83061783
Errore standard	1.59108357
Osservazioni	8

ANALISI VARIANZA

	<i>gdl</i>	<i>SQ</i>	<i>MQ</i>	<i>F</i>	<i>Significatività F</i>
Regressione	2	6510.621406	3255.31	1285.897839	1.65854E-07
Residuo	6	15.18928147	2.53155		
Totale	8	6525.810688			

$$T = C(1) \times n + C(2) \times n^2$$

	<i>Coefficienti</i>	<i>Errore standard</i>	<i>Stat t</i>	<i>Valore di significatività</i>	<i>Inferiore 95%</i>	<i>Superiore 95%</i>
C(1)	0.02897293	0.004418372	6.55738	0.000602233	0.018161552	0.039784302
C(2)	8.8199E-05	6.29393E-06	14.0134	8.23612E-06	7.27985E-05	0.0001036

Dalla regressione risulta che il coefficiente di  $n^2$  influenza la variabile dipendente  $T$  (tempo di calcolo) infatti il suo p-value è 0.0000, inoltre dividendo il coefficiente per il suo errore standard si ottiene una **t di Student** che testa l'ipotesi nulla ( $H_0$ ) che il coefficiente sia uguale a zero.

Possiamo quindi concludere che il nostro algoritmo appartiene all'ordine del tempo di calcolo quadratico  $O(n^2)$ .

Applicando ora, a questi ultimi algoritmi, una matrice di dati reali c'accorgiamo che c'è ancora un piccolo problema da risolvere, ossia ogni descrizione è formata da elementi che possono essere lettere maiuscole, lettere minuscole, simboli e numeri, e le dimensioni di ognuno di essi sono diverse, quindi dobbiamo trovare un tipo di carattere per cui ogni elemento ha la stessa dimensione, questo problema è risolto utilizzando il carattere "Andale Mono" come possiamo vedere dal risultato finale.

	0
X2 =	" DERIVATI SU CAMBI OTC CP _____"
	" Contatto (derivati su cambi OTC) _____"
	" RICHIEDERE la verifica delle linee di credito _____"
	" Mancato rispetto disposizioni interne: mancata verifica censimento ctp (a) ___"
	" Esercizio opzioni e gestione scadenze futures (derivati su cambi OTC) _____"
	" INSERIRE i dati relativi all esercizio su MxG2000 _____"
	" Mancato esercizio di un opzione in the money _____"
	" Invio conferme (derivati su cambi OTC) _____"
	" VERIFICARE la conferma di MxG2000 _____"
	" Errato input manuale nei sistemi (correz. conferma) _____"
	" Input e Validazione (derivati su cambi OTC- Spot/Forward) _____"
	" VALIDARE l operazione in MxG2000 _____"
	" Mancata validazione dei deals nei sistemi di FO _____"
	" INSERIRE i dati relativi all operazione nel foglio Excel _____"
	" Errato input manuale (parametri operazione) _____"
	" Ricezione flussi contabili (MXRATES) 15 _____"
	" ATTIVARE la query di investigation in PERRY MASON 15 _____"
	" Inefficienza procedurale: mancata identificazione resp. risoluzione errore 15___"
	" ATTIVARE le procedure informatiche per l acquisizione dei dati in RECON* 15 _____"

Ora possiamo affermare che il nostro algoritmo è finito; nella riga zero possiamo vedere il Top Event e di seguito tutta la scala gerarchica.

## ***BIBLIO – WEBGRAFIA***

- *STRUTTURA, ORGANIZZAZIONE E PROGETTO DEI CALCOLATORI*  
David A. Patterson, John L. Hennessy con un contributo di James R. Larus  
Casa Editrice: JACKSON LIBRI
  
- *DATA STRUCTURES AND ALGORITHMS IN JAVA*  
Michael T. Goodrich, Roberto Tamassia  
Casa Editrice: Jhon Wiley & Sons
  
- *IL RISCHIO FINANZIARIO*  
Umberto Cherubini, Giovanni Della Lunga  
Casa Editrice: Mc. Grow-Hill
  
- Riferimenti da: *COLLECTION AND PRELIMINARY ANALYSIS OF FAILURE DATA* Documento consultivo
  
- Riferimenti da: *MODELIN AND ANALYSIS OF MULTI-COMPONENT SYSTEM* Documento consultivo
  
- [www.eng.it/](http://www.eng.it/) - 2k
- [www.engineering.it/](http://www.engineering.it/) - 18k - 22 set 2004
- [www.di.unipi.it/~andrea/Didattica/SP-LSD-04/LSD/AlberiBinari/](http://www.di.unipi.it/~andrea/Didattica/SP-LSD-04/LSD/AlberiBinari/)
- [www.di.unito.it/~liliana/PROGRAMMAZIONE/Alberi1.ppt](http://www.di.unito.it/~liliana/PROGRAMMAZIONE/Alberi1.ppt)
- [www.di.unipi.it/~andrea/Didattica/SP-LSD-04/LSD/AlberiBinari/](http://www.di.unipi.it/~andrea/Didattica/SP-LSD-04/LSD/AlberiBinari/) - 2k
- [web.ing.unisannio.it/canfora/pdf/searchtree.pdf](http://web.ing.unisannio.it/canfora/pdf/searchtree.pdf)
- [staff.polito.it/gianpiero.cabodi/dida/apa/teoria/cap-5.5-bst.PDF](http://staff.polito.it/gianpiero.cabodi/dida/apa/teoria/cap-5.5-bst.PDF)
- [www.dis.uniroma1.it/~liberato/tecniche/nalberi/](http://www.dis.uniroma1.it/~liberato/tecniche/nalberi/) - 2k
- [www.dia.uniroma3.it/~lambda/asd/](http://www.dia.uniroma3.it/~lambda/asd/) - 3k

- [www.dis.uniroma1.it/~becchett/algo/slide/parte4.ppt](http://www.dis.uniroma1.it/~becchett/algo/slide/parte4.ppt)
- [gauguin.info.uniroma2.it/~finocchi/INFOII2004/slides/lecture10.ppt](http://gauguin.info.uniroma2.it/~finocchi/INFOII2004/slides/lecture10.ppt)
- [www.dm.unibo.it/matematica/AlgebraLineare/diz1/ordine.htm](http://www.dm.unibo.it/matematica/AlgebraLineare/diz1/ordine.htm) - 5k
- [www.dimat.unina2.it/mazzocca/algebra1/capit4.htm](http://www.dimat.unina2.it/mazzocca/algebra1/capit4.htm) - 56k
- [www.batmath.it/matematica/a\\_relazioni/ordine.htm](http://www.batmath.it/matematica/a_relazioni/ordine.htm) - 6k
- [www.science.unitn.it/~moretti/App.pdf](http://www.science.unitn.it/~moretti/App.pdf)
- [gs.ing.univaq.it/~gabriele/Scintillae/portelog.htm](http://gs.ing.univaq.it/~gabriele/Scintillae/portelog.htm) - 6k
- [it.wikipedia.org/wiki/Algebra\\_booleana](http://it.wikipedia.org/wiki/Algebra_booleana) - 46k
- [www.mfn.unipmn.it/~bobbio/DIDATTICA/ARCH1\\_00/ALDISP\\_00pdf](http://www.mfn.unipmn.it/~bobbio/DIDATTICA/ARCH1_00/ALDISP_00pdf)
- [www.basilea2.com/basilea\\_2.htm](http://www.basilea2.com/basilea_2.htm) - 28k
- [www.studiobottero.it/s\\_basilea2.asp](http://www.studiobottero.it/s_basilea2.asp) - 25k
- [www.tidona.com/pubblicazioni/maggio03\\_4.htm](http://www.tidona.com/pubblicazioni/maggio03_4.htm) - 50k
- [www.bancariaeditrice.it/portal/ssm/page.do?pageId=10774](http://www.bancariaeditrice.it/portal/ssm/page.do?pageId=10774) - 98k
- [www.capitalia.it/pages/basilea2\\_txt.htm](http://www.capitalia.it/pages/basilea2_txt.htm) - 8k
- [www.adfor.it/consulting/basilea2rischioop.asp](http://www.adfor.it/consulting/basilea2rischioop.asp) - 28k